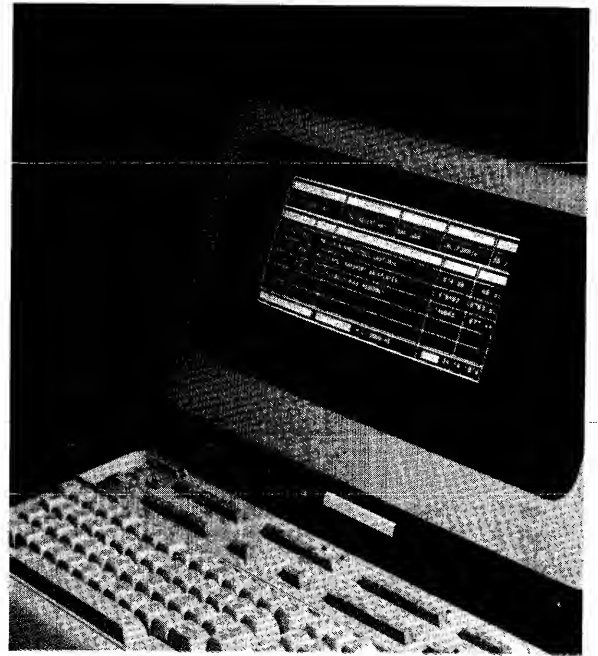
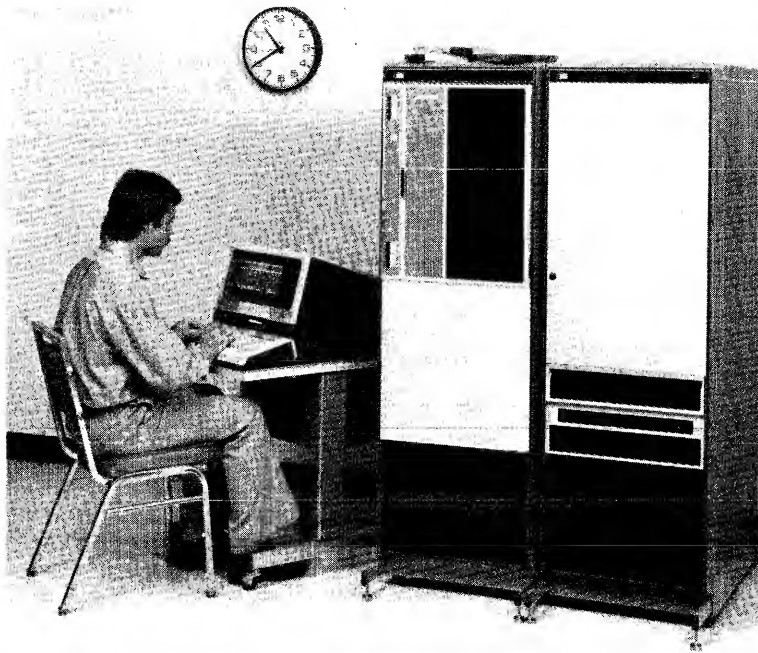


RTE II/III: A Guide for New Users



RTE II/III: A Guide for New Users



HEWLETT-PACKARD COMPANY
11000 WOLFE ROAD, CUPERTINO, CALIFORNIA, 95014

PUBLICATION NOTICE

Changes in text to document software updates subsequent to the initial release are supplied in manual change notices and/or complete revisions to the manual. The history of any changes to this edition of the manual is given below under "Publication History." The last change itemized reflects the software currently documented in the manual.

Any changed pages supplied in an update package are identified by a change number adjacent to the page number. Changed information is specifically identified by a vertical line (change bar) on the outer margin of the page.

PUBLICATION HISTORY

Third Edition Aug 78

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another program language without the prior written consent of Hewlett-Packard Company.

Copyright © 1978 by HEWLETT-PACKARD COMPANY

Library Index No.
2RTE.310.92060.90012

PREFACE

This manual, an introduction to using the RTE-III operating system, presents the fundamentals of RTE-III. Use the manual as a workbook, trying each example as you read the explanation of the system features.

- Section I describes some fundamental characteristics of the system and introduces basic operating skills.
- Section II is a step by step description of compiling and running a FORTRAN IV program.
- Section III outlines other ways to prepare programs.
- Section IV provides tips on diagnosing problematic situations.

Before using this manual, read the RTE-III General Information Manual to familiarize yourself with the concepts and capabilities of the system. In addition, the GIM defines some terminology which is used in this manual.

This manual assumes you have had prior experience programming in FORTRAN IV. You should have the RTE FORTRAN IV Reference Manual handy while you are doing the exercises in order to check compilation errors and determine correct statement formats. If you want to use the system for BASIC programming, only Section I and parts of Sections III and IV will be useful to you.

As you study the manual it will occasionally recommend that you consult with the system manager. System manager is our term for the most experienced person at your installation. Such a person will more than likely have taken the HP RTE training course or gained experience from extensive use of the system. If you are the first person to use the system at your installation your system manager is your local HP Customer Engineer.

The instructions in this manual are intended for use at the system console in a single user environment. Special instructions are given when necessary for users in a Multi-Terminal Monitor (MTM) environment. However, the system and user interaction in all of the examples is shown only as it appears on the system console.

FOR RTE-II USERS

The term RTE always refers to RTE-III because the instructions in this manual are specifically for that system. However, the manual may be useful to you if you have an RTE-II system and the File Management Package. The only differences are:

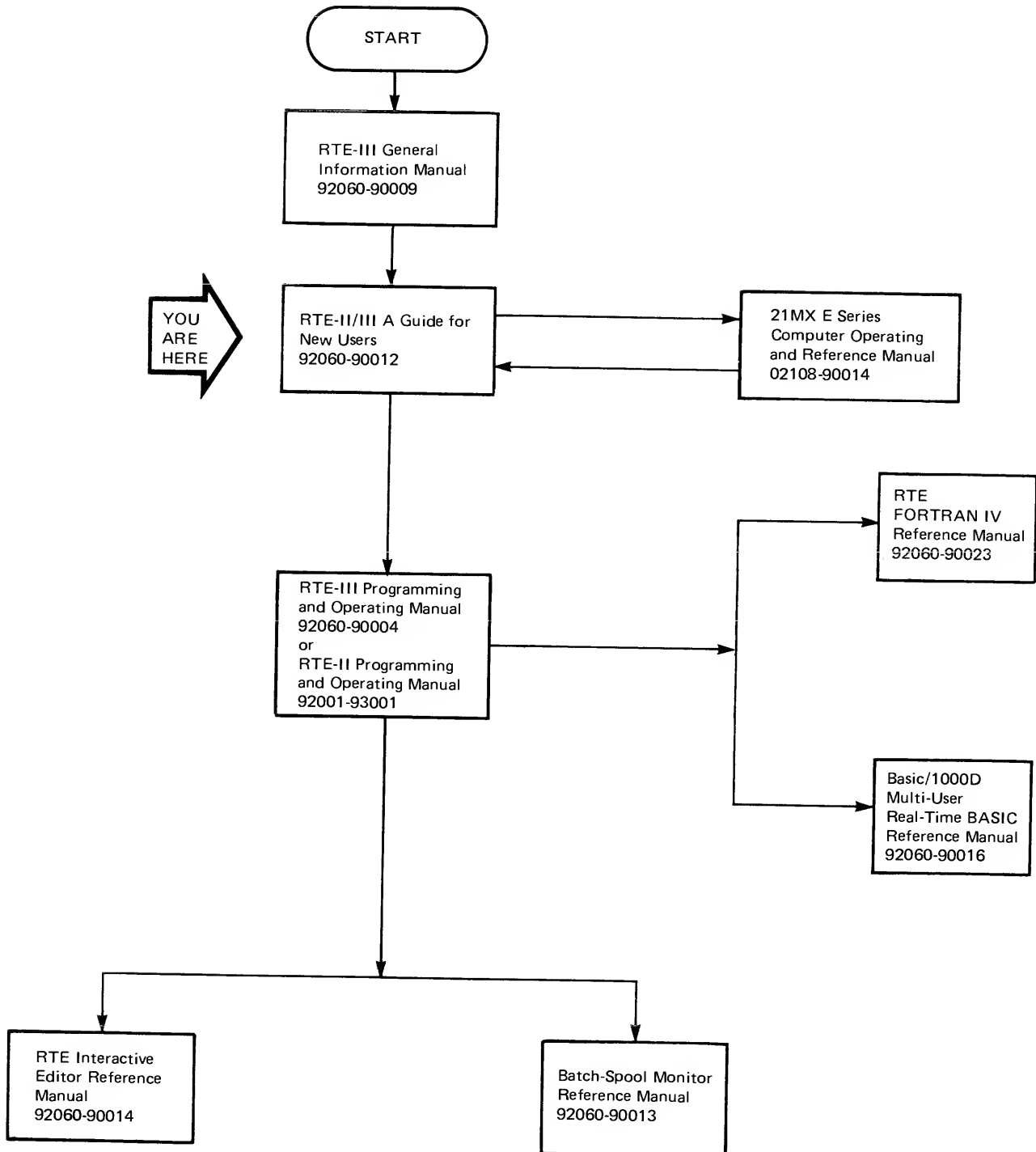
- the procedure for loading the system,
- the information provided when you use the STATUS command,
- occasional references to partitions that do not apply to RTE-II.

Because this is an introductory manual, only a small subset of the many commands and options available with RTE are described. When you have finished this manual, you can study the following manuals to learn about the more powerful and flexible features of the system:

- RTE-III and RTE-II Programming and Operating Manuals
- Batch-Spool Monitor Reference Manual
- RTE Interactive Editor Reference Manual

See the documentation map for a diagram of RTE manuals and their relation to each other.

DOCUMENTATION MAP



CONTENTS

PREFACE	iii
DOCUMENTATION MAP	v

SECTION I	FUNDAMENTALS	1-1
	Turning on the Equipment	1-2
	Loading the RTE Software	1-5
	The System Console	1-8
	Input/Output Devices	1-9
	RTE Commands	1-10
	Request the Time	1-10
	Set the Real-Time Clock	1-10
	Request Program Status	1-11
	Schedule Programs	1-11
	Device Problems	1-12
	Terminate Programs	1-12
	Using the File Manager	1-13
	FMGR Error Messages	1-14
	File Names	1-14
	Create a File	1-16
	List the Content of a File	1-17
	List information on the System Console	1-18
	The Editor	1-19
	More FMGR Commands	1-24
	Program Development	1-26
	ID Segments	1-27
SECTION II	Program Preparation	2-1
	Step 1. . .Create a Source File	2-2
	Step 2. . .Edit a Source File	2-6
	Step 3. . .Compile a Program	2-8
	Step 4. . .Load and Run a Program	2-11
	Step 5. . .Correct Program Logic Errors	2-15
	A Summary	2-16
SECTION III	RTE FEATURES	3-1
	Multi-Terminal Monitor	3-2
	BASIC Programs	3-4
	Batch Job Control	3-6
	Error and Other Messages	3-9
SECTION IV	TROUBLESHOOTING	4-1
	Program Status	4-1
	The WHZAT Program	4-3
	Device Problems	4-6
	Symptoms and Remedies	4-7
APPENDIX A	COPYING PROGRAMS	A-1

ILLUSTRATIONS

System Switches	1-4
The HP 1000 Computer Front Panel	1-7
Typical RTE Terminal	1-8
Example of Creating a Source File	2-3
FORTRAN Program Listing	2-5
Example of Editing a Source File	2-6
Compile FORTRAN IV Program	2-8
FORTRAN Listing	2-10
Loading and Running a Program	2-11
LOADR Listing	2-13
Using a Multi-Terminal Device	3-3
Creating a Program Using BASIC Interpreter	3-5
Creating a BASIC program with EDITR	3-5
Job Deck Containing FMGR Commands	3-7
Preparing a Procedure File Containing a Job	3-7
Program Status Mode (*RUN,WHZAT)	4-4
Partition Status Mode (*RUN,WHZAT,,1)	4-5

TABLES

Error Messages, Step 1	2-4
Error Messages, Step 2	2-7
Error Messages, Step 3	2-9
Error Messages, Step 4	2-12
Termination and Error Messages	3-9
Program States	4-2

FUNDAMENTALS

SECTION

I

Before you begin running programs under RTE there are several things you need to know. This section contains information to help you perform the tasks in Sections II and III. You will need to understand:

- The concept of logical unit numbers.
- How to start the system: turn on the hardware and load the RTE software.
- How to use the system console or a terminal.
- How to prepare and save program files using the File Manager and Interactive Editor.

If your system is already operating, you may want to skip the step-by-step start up procedure and proceed to the system console instructions and information about commands and files. You can then turn to Section II and do the exercise described there.

In order to provide consistent instructions and examples in this manual, all the discussion is related to an RTE system with the following characteristics:

- The system console is an HP interactive display terminal with two mini-cartridge tape units.
- The standard output unit is one of the system console's mini-cartridge tape units.
- The standard list unit is a line printer.
- The standard input unit is one of the system console's mini-cartridge tape units.
- The system disc is a subchannel of an HP disc drive.
- The system has ROM bootstrap loaders.

Occasionally other types of devices are described but examples of actual interaction conform to these standards.

If the RTE system you are using is not exactly like the one described here, you will still be able to use this manual but you must translate terms occasionally to adapt the instructions to your computer configuration. Since most of the discussion is in terms of logical unit numbers, this should not be a problem.

Turning on the Equipment

If your system is already running, you can skip this description and go to the next topic. If not, here is a set of step-by-step instructions for turning on the equipment and starting the RTE system.

The equipment illustrated may not look exactly like yours but you can find the comparable buttons and devices by carefully comparing figure 1-1 and your equipment. Devices which are not required by the tasks in this manual are not described and you can ignore them for now. If you cannot find everything, ask your system manager to help you.

① Turn on Disc Drive Power

If your system is a vertically racked model, go to the front of the cabinet and locate the system ON/OFF power switch in the upper right corner of the rack, as shown in Figure 1. Press this switch, which will light up.

If your system is a desk model, go to the rear of the disc drive unit. Locate the power switch in lower left corner as shown in Figure 1 and set it to ON.

② Turn on Main Power to Computer

If you have a vertically racked model, you turned on the main power to the computer when you set the system ON/OFF switch to ON. Proceed to Step 3.

If you have a desk model, locate the POWER switch at the rear of the computer mini-rack in the bottom left corner, as in Figure 1. Set the POWER switch to ON.

③ Insert Disc Cartridge

If you want to change or insert a disc cartridge, first set the RUN/STOP switch (or the LOAD/UNLOAD switch if you have a 7900 disc subsystem) on the front of the disc drive (see Figure 1) to STOP (UNLOAD). When the DOOR UNLOCKED light comes on, open the door of the drive, pulling the top of the door firmly out towards you, and pull out the cartridge. Insert a new cartridge by grasping the handle of the cartridge, and with the labeled surface up, slide the disc in and close the door.

④ Set RUN/STOP (LOAD/UNLOAD) Switch

On the front of the disc drive cabinet, locate the RUN/STOP (LOAD/UNLOAD) switch. Set it to RUN (LOAD) and observe that the door unlocked light goes out.

⑤ Verify Correct System Start-Up

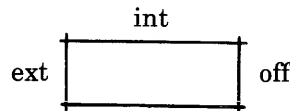
"TERMINAL READY" should be displayed at the top of the screen of the system console. Look at the DISC READY indicator on the front of the disc drive unit. It will take about 30 seconds for the disc to come up to speed, and the DRIVE READY indicator will light.

If you have no display on the system console:

- a. Check to make sure the terminal is plugged in and that the power switch on the rear of the terminal under the rear door is Set to ON.
- b. Perform the SHUT DOWN procedure and then wait 5-10 seconds.
- c. Perform the Start-up procedure again.

⑥ Turn on Power Fail Battery

Now that the system is running, locate the battery box inside the rear of the computer cabinet. Find the battery switch marked



behind the battery cable as shown at the top of Figure 1. Set this switch to int. This turns the power-fail battery ON, which protects the main memory of the computer for up to two hours if power should fail. When power returns, the computer will then be able to resume operation with no loss of memory.

Occasionally other types of devices are described but examples of actual interaction conform to these standards.

If the RTE system you are using is not exactly like the one described here, you will still be able to use this manual but you must translate terms occasionally to adapt the instructions to your computer configuration. Since most of the discussion is in terms of logical unit numbers, this should not be a problem.

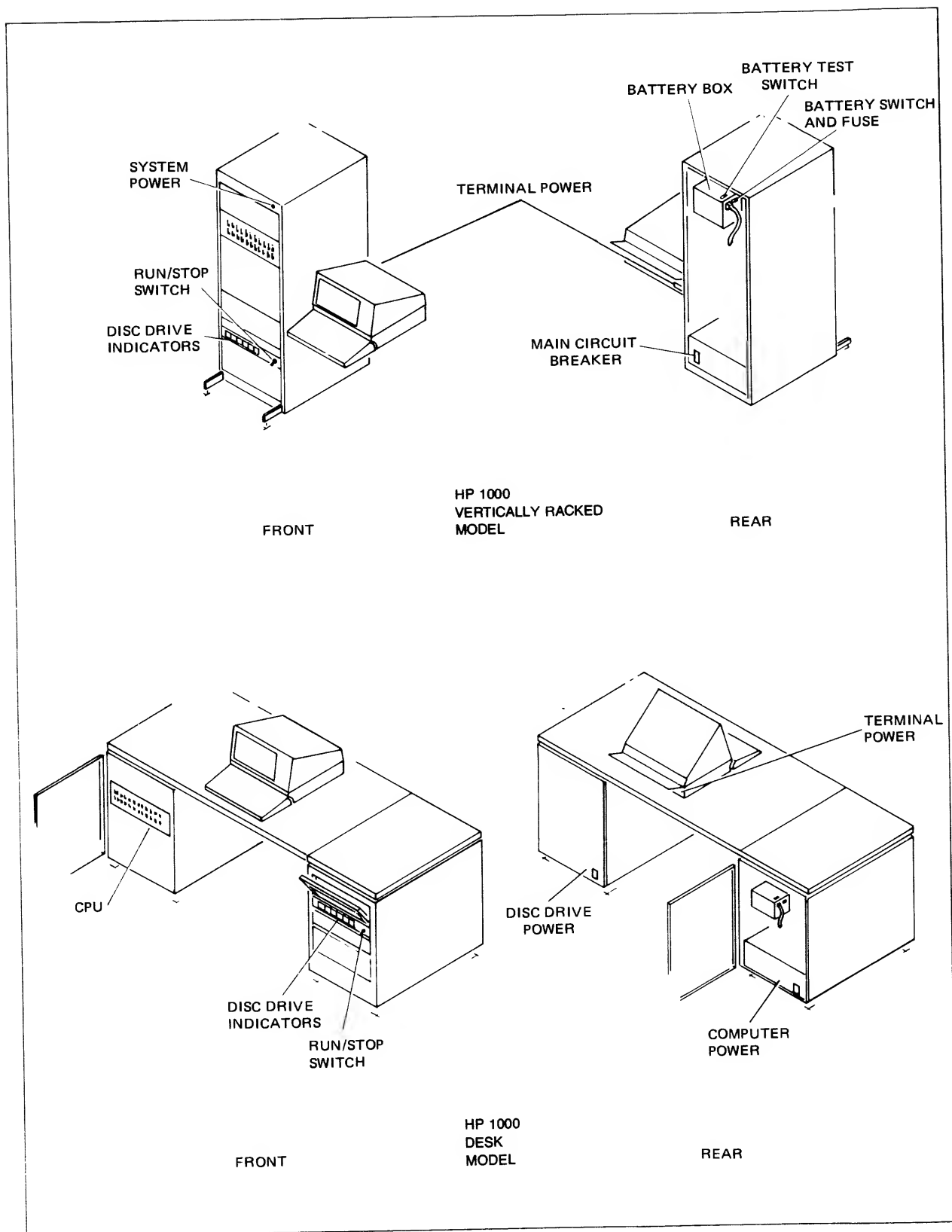


Figure 1-1. System Switches

Loading the RTE Software

Before you can load the RTE system software into memory from the system disc, you should be familiar with the HP 1000 computer front panel. Figure 1-2 contains a detailed drawing of the panel with an explanation of various buttons, switches and lights labeled with letters. If you are already familiar with the front panel, skip the explanation and go directly to the load instructions which refer to the letters in figure 1-2.

It is not necessary to completely understand the function of all these components to load the RTE system. You will become familiar with them as you use them. For now, you only need to know enough to load the bootstrap loader which loads the system software. If you want more details about the front panel, read the HP 1000 Computer Operator's Manual.

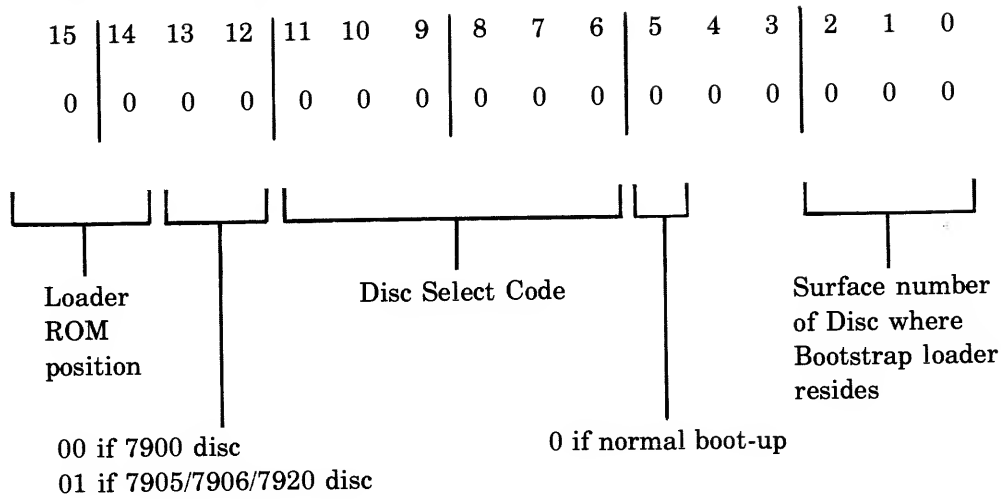
Note: The EXTEND, PARITY, and INTERRUPT SYSTEM lights can be ignored for now unless PARITY persistently lights. In that case, call the system manager. The INC M/DEC M switch is used to increment the content of the M register. It need not be selected now.

The procedure below assumes that the system power and all devices are on, as described in chapter 1, "Turning on the Equipment".

To start up your HP 1000 manually, follow this procedure.

1. Open the front door of the cabinet containing the computer with the key provided.
2. Open the front panel of the CPU with the key provided and set the LOCK/OPERATE switch inside to OPERATE.
3. Close the front panel.
4. Press the HALT switch.
5. Display the S register by pressing the Register Select switch until the indicator light is over the S label.
6. Press the Clear Display switch.

7. Enter the following numbers into the Display Register



Loader ROM position: see the Loader ROM Installation Manual for the code.

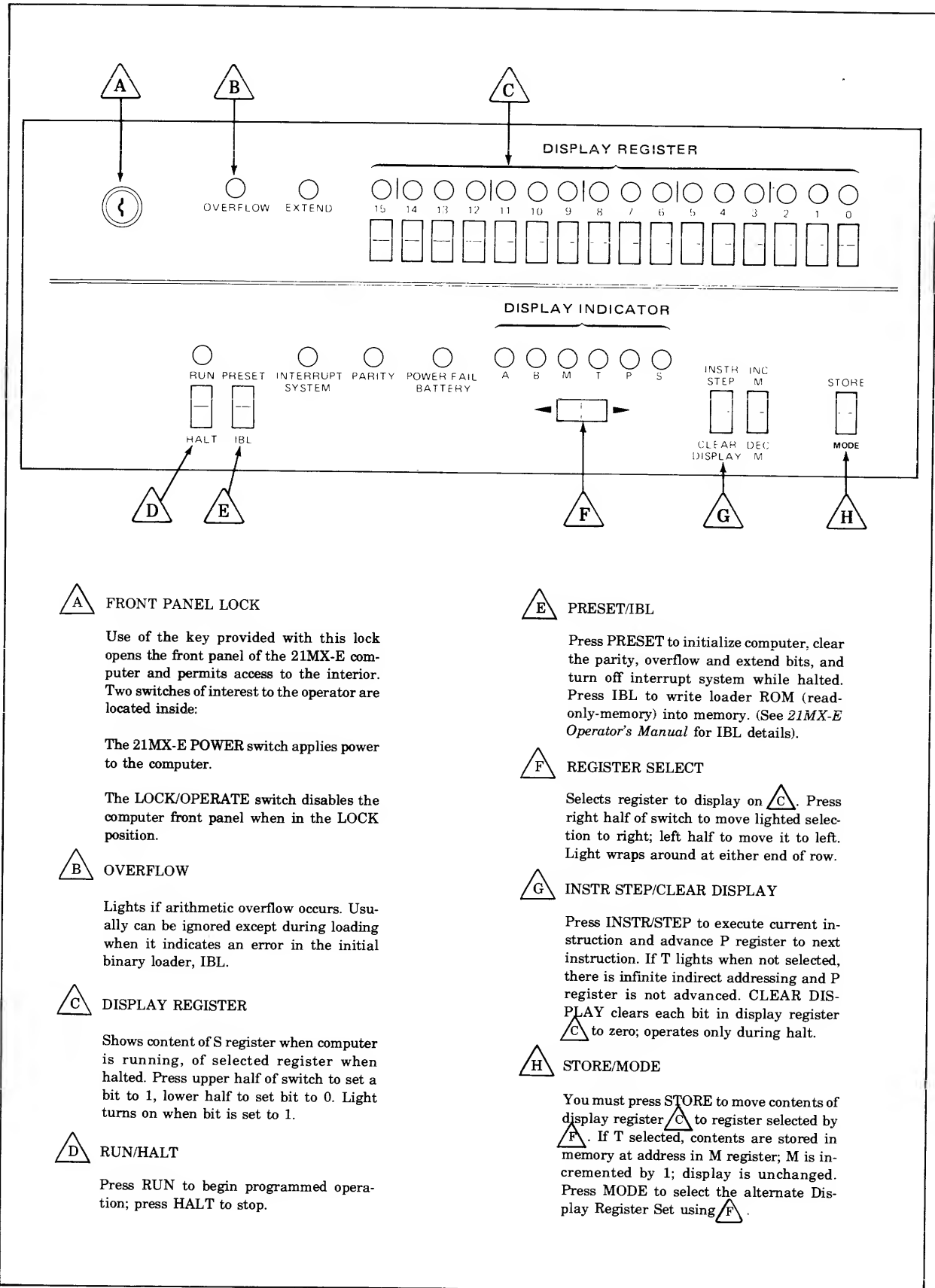
Disc Select Code: see the Primary System Hardware Configuration table in the System Support Log in the system documentation.

8. Press the STORE switch.

9. Press the PRESET, IBL and then the RUN switches.

10. If bit 5 was NOT set, and the procedure has been followed correctly, your RTE System is now running.

RTE IS RUNNING!



A FRONT PANEL LOCK

Use of the key provided with this lock opens the front panel of the 21MX-E computer and permits access to the interior. Two switches of interest to the operator are located inside:

The 21MX-E POWER switch applies power to the computer.

The LOCK/OPERATE switch disables the computer front panel when in the LOCK position.

B OVERFLOW

Lights if arithmetic overflow occurs. Usually can be ignored except during loading when it indicates an error in the initial binary loader, IBL.

C DISPLAY REGISTER

Shows content of S register when computer is running, of selected register when halted. Press upper half of switch to set a bit to 1, lower half to set bit to 0. Light turns on when bit is set to 1.

D RUN/HALT

Press RUN to begin programmed operation; press HALT to stop.

E PRESET/IBL

Press PRESET to initialize computer, clear the parity, overflow and extend bits, and turn off interrupt system while halted. Press IBL to write loader ROM (read-only-memory) into memory. (See 21MX-E Operator's Manual for IBL details).

F REGISTER SELECT

Selects register to display on **C**. Press right half of switch to move lighted selection to right; left half to move it to left. Light wraps around at either end of row.

G INSTR STEP/CLEAR DISPLAY

Press INSTR/STEP to execute current instruction and advance P register to next instruction. If T lights when not selected, there is infinite indirect addressing and P register is not advanced. CLEAR DISPLAY clears each bit in display register **C** to zero; operates only during halt.

H STORE/MODE

You must press STORE to move contents of display register **C** to register selected by **F**. If T selected, contents are stored in memory at address in M register; M is incremented by 1; display is unchanged. Press MODE to select the alternate Display Register Set using **F**.

Figure 1-2. The HP 1000 Computer Front Panel

The System Console

Look at the system console. RTE has typed the message:

SET TIME

Now you may set the real-time clock to the current time or go ahead and use other RTE commands. However, before using the commands you should be familiar with some HP interactive display console features. Figure 1-3 shows a terminal often used as a system console, the HP 2645 Interactive Display Console. Make sure you locate the following special keys which you will be using frequently:

RETURN (carriage return) indicates the end of the current input record.

DEL deletes the line you just typed.

BACKSPACE deletes a character.

When using the terminal to communicate to the system, you should have both the REMOTE key and CAPS LOCK key in the down position. The AUTO LF key should be in the up position.

Now turn to the next topic and start using RTE.

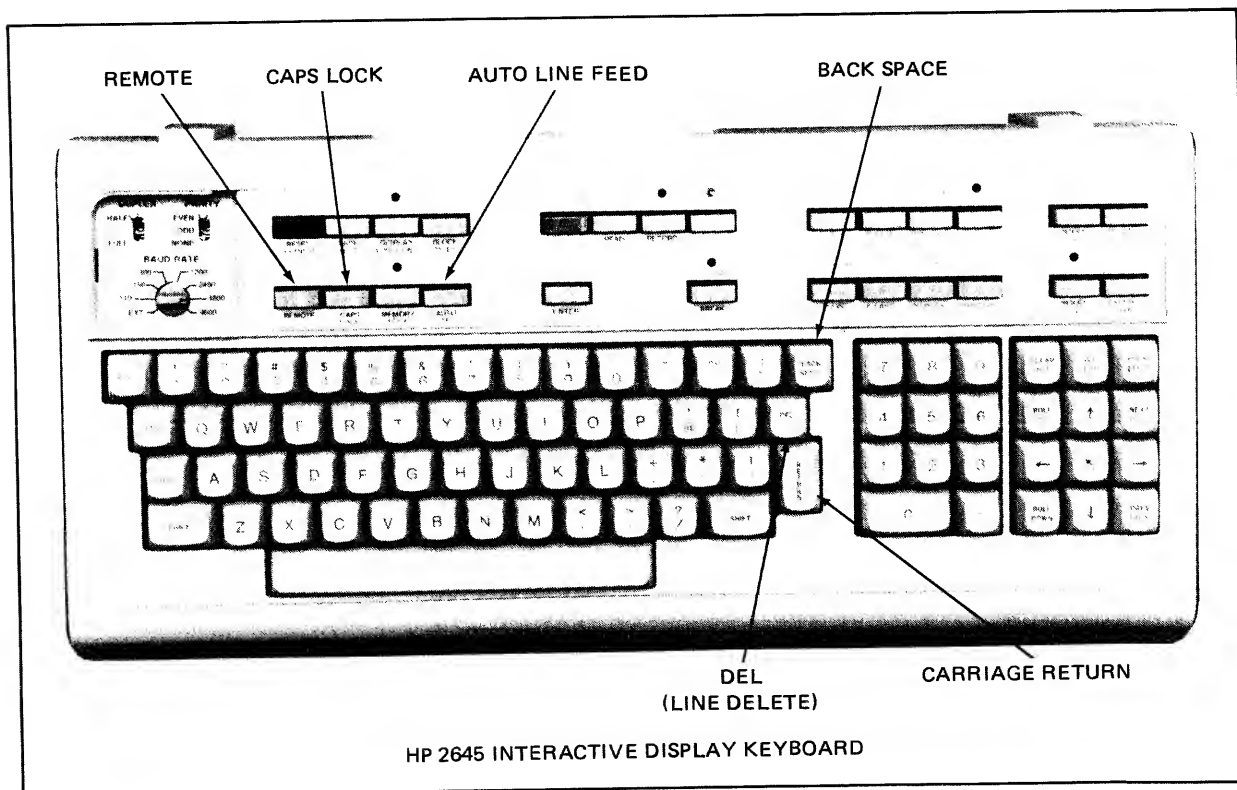


Figure 1-3. Typical RTE Terminal

Input/Output Devices

Before you begin experimenting with RTE you must know about logical unit numbers. Input and output devices are assigned logical unit numbers when the RTE system is generated and you then use these numbers to address all devices. Many devices such as line printers are addressed by one logical unit number while others such as disc drives have subchannels with each subchannel addressed by a different logical unit number.

Logical unit numbers, abbreviated LU in this manual, are decimal integers between 0 and 63. Numbers 1 through 6 must always refer to the following devices:

- 1 - system console
- 2 - system disc
- 3 - auxiliary disc (optional)
- 4 - standard output unit
- 5 - standard input unit
- 6 - standard list unit.

The standard output unit is a mini-cartridge unit or magnetic tape drive. The standard input unit is usually a mini-cartridge unit but may be a card reader or terminal. The standard list unit may be a line printer or a terminal. The remaining logical unit numbers (7-63) may be assigned to any type device although it is recommended that you use 8 for magnetic tape since File Manager device control commands default to LU 8. File Manager assumes LU 8 is magnetic tape unless another type device is specified.

Logical unit number 0 is not associated with a particular device, it has a special purpose which is described in the RTE Programming and Operating Manual.

Although a device or subchannel of a device may be assigned more than one logical unit number, those shown in this manual have only one. The system and auxiliary discs may only be assigned the single LU numbers 2 and 3 respectively. The logical unit number in a FORTRAN input/output statement and the unit number in an ALGOL input/output statement correspond to the LU number discussed here. For example in a FORTRAN print statement such as:

```
WRITE (1,200)
200 FORMAT ("HELLO")
```

the data will be printed on LU 1, the system console.

If you write an Assembly language program, you use EXEC calls to instruct the RTE system to perform input/output operations. These calls are described in the General Information Manual. They are not discussed here except to point out that the LU number is specified as part of the control word parameter of any EXEC calls that perform input/output.

You also use logical unit numbers in operator commands as you will see a little later in this section.

One other important thing you should know about I/O devices is that RTE allows you to specify a timeout value for a device. This is done at the time the system is generated (or with an operator command) and defines how long the system will wait for an indication that an I/O operation is complete before declaring the device down (inoperative). For now, you only need to understand that there may be a delay before you are notified that a device such as the line printer is down and you see a message to that effect printed on the system console.

RTE Commands

What are RTE commands and what are the format conventions? You, the operator, use RTE commands to control the RTE system and request RTE to perform a variety of functions. Specific commands will be described as they are used in the tasks which follow. However, you should follow certain conventions with all system commands:

- You may type the full command name but only the first two characters are required.
- All blanks are ignored.
- The number and type of parameters depends on the particular command.
- Separate the command and parameters with commas.
- Terminate each command and parameter list by pressing RETURN to enter the command.
- Use uppercase for all commands and alphabetic parameters.

REMEMBER. . .if you make a mistake you can correct it by typing BACKSPACE, or you can delete the whole line by pressing DEL.

First you must indicate to the system that you want to do something. Press any key on the system console. RTE will respond with an asterisk (*), the RTE prompt character, indicating it is ready to accept your command. You must do this each time you want to enter a command. If you are using a MTM terminal instead of the system console, the prompt is your terminal LU number followed by a greater than sign, >, for example, 07>. In either case, you are now in control and can enter a command.

*HI

OP CODE ERR

Try typing HI followed by RETURN. RTE responds with an error message because it does not understand the command HI. The point of this exercise is to show you that nothing drastic occurs if you make a mistake so don't be timid about trying the exercises which follow.

Request the Time

The TIME command instructs RTE to print the year, day, and time according to the system real-time clock. This is a good command to try first because it is very straightforward and does not have any parameters.

*TIME
1975 91 14 6 59
↑ ↑ ↑ ↑ ↑
year julian day hour minutes
seconds

PRESS any key. RTE prints the prompt. Type the letters TIME and press RETURN. The system prints the *year*, *julian day*, *hours*, *minutes* and *seconds* currently recorded in the real-time clock. (The *julian day* is a serial number equal to the number of days elapsed since January 1.)

If you make a mistake and the message OP CODE ERROR is printed, simply enter the command again.

Set the Real-time Clock

If you loaded the system yourself, the time will probably be incorrect because you have not yet been instructed by this manual to set the time. The system automatically sets the real-time clock to 8 a.m. and a default date. You may set the clock to the correct time with the TM command. This is the normal procedure after receiving the SET TIME message when starting the system. If you did not load the

system, you should be very careful with the TM command (and probably not use it until you know all the intricacies of the system). Some real-time processing may be dependent on the clock being accurately set.

To set the clock, press any key. When the asterisk appears, type TM and the parameters as shown.

<i>year</i> ↓ <i>hour</i> ↓ <i>seconds</i> ↓ *TM,1975,115,15,25,15 ↑ ↑ <i>julian day</i> <i>minutes</i>	Type the year (4 digits), julian day (3 digits), hour (2 digits based on a 24 hour clock), minutes (2 digits), and seconds (2 digits).
--	--

When you press RETURN, the clock is set. In this example, the time will be April 25, 1975, 3:25:15 p.m. The *hour*, *minutes*, and *seconds* parameters are optional and will be set to 0 if omitted.

Request Program Status

Another command which may be useful to you later is the STATUS command that requests the system to report on the status of a program. STATUS has one parameter. You may omit this parameter or set it to 0 to determine which program is currently executing, set it to a program name to get the status of that program, or set it to a partition number to get the name of the program currently in that partition. Partitions are described in the General Information Manual.

*STATUS Press any key and then type ST or STATUS followed by RETURN.
 PROGX 23

If the system displays a zero, no program is currently executing. If the system replies with anything other than a standard error message, it will be the name of a program which is currently executing and the partition it currently resides in. In the example, PROGX is executing in partition 23.

You may type STATUS, a comma, and a partition number to request the name of a program in that partition. If the system replies with a 0, there is no program in that partition; or if NO SUCH PROG, there is no such partition.

*STATUS,23 Type the command and partition number 23. The program name in that partition is printed.
 PROGX

In the next section you will be using the program FMGR. You may check the status of FMGR by typing STATUS,FMGR. Remember to press a console key first and the RETURN at the end.

*STATUS,FMGR

50	0	0	0	0	0	0	0
↑	↑	└──────────┘				↑	
priority	program state	clock values					

The system replies with some information about FMGR. The first number is the *priority* assigned to the program. The second number indicates the current state of the program. (Zero means the program is dormant, waiting to be executed.) The remaining numbers are values such as hours, minutes, and seconds used to schedule the program by the clock. FMGR is not normally scheduled that way so the values are zeroes.

More information is provided in Section IV about the status of the programs. A full explanation of the clock values is contained in the RTE Programming and Operating Manual.

Schedule Programs

The RUN command is used to schedule programs from the system console. You must type characters RUN (or RU) followed by the name of the program you want to run.

*RUN,FMGR	The colon is FMGR's prompt character, its way of indicating it is ready to
:EX	accept a File Manager command. FMGR commands are the next topic dis-
\$END FMGR	cussed in this section. For now you can type EX to terminate the File Manager.

If you make a mistake entering the command, the following error messages may be printed:

NO SUCH PROG	You probably misspelled the program name. Try again.
OP CODE ERR	Did you type the command correctly?
ILLEGAL STATUS	The program is already running and may be in a suspended state. Ask the system manager for help. (If you are operating on a MTM terminal, do not try the command again since someone else is probably using FMGR. Just continue with the next topic.)

Device Problems

If an input/output device is down, you may need to use the UP command to notify RTE after you have diagnosed and fixed the problem. When the device goes down RTE prints one of the following errors:

I/O ERR TO EQT <i>#eqt</i>	The device has timed out (it is down).
I/O ERR NR EQT <i>#eqt</i>	The device is not ready.
I/O ERR ET EQT <i>#eqt</i>	An end of tape has been encountered.

Each device has a number associated with it which is called its *EQT entry number*. EQT stands for Equipment Table, a table maintained by the system to describe the I/O equipment and identify the driver used for each device. The EQT number represents the number of a device's entry in this table. You will only need to use this number in the tasks described in this manual if a device is down.

When the device is ready after any of the problems described above have been corrected, type:

*UP,*eqt* where *eqt* is the same number typed in the I/O error message.

If you do not know which device the equipment number refers to, you can check the devices you are using or ask the system manager.

Terminate Programs

If you want to terminate a program you can use the OFF command. You can type the command followed by the program name. For example:

*OFF,PROGX

The program will be made dormant in an orderly way by allowing any current I/O operation to complete before terminating.

If you want the program to terminate immediately, you can type a 1 as the parameter following the program name.

*OFF,PROGX,1 You should use this parameter value carefully so you do not abort an I/O operation such as reformatting a disc. Section IV discusses this command more specifically.

A third form of this command (*OFF,PROGX,8) is discussed with ID segments at the end of this section.

Using the File Manager

The File Manager (which is part of the Batch-Spool Monitor) provides commands which, among other things, make it easy and convenient for you to create programs and save them. It also provides simple commands for doing other file operations such as copying and purging. This manual will not show you how to use all the File Manager capability but will show you just enough to do some simple tasks. The Batch-Spool Monitor Reference Manual describes all the commands.

You should remember that the File Manager is not part of the system. It is just another background program which may be swapped out in favor of a higher priority program. As a result, other programs running in the system may print information on your terminal. In addition, if you are using the system console, the system may report error information it detects on other programs. In most cases you can ignore the messages but occasionally you may need to ask the system manager what to do.

The File Manager command conventions are almost identical to the RTE conventions you have been using. They are the same in that each command consists of at least two characters and an optional list of parameters separated by commas. They differ in that a parameter is sometimes divided into subparameters separated by colons. File Manager commands, like RTE commands, are transmitted by pressing RETURN. Unlike the system, you do not have to hit a key to get the attention of the File Manager. It will type the prompt character as soon as it is ready to accept the next command. For File Manager, the prompt character is a colon(:) rather than an asterisk (*) or LU >.

In addition to file management, the Batch-Spool Monitor has commands to control batch job processing and spooling of input/output operations. The spooling feature is beyond the scope of this manual but batch job control is described briefly in Section III. Both features are described thoroughly in the Batch-Spool Monitor Reference Manual.

It is important to know whether your system is running batch jobs since the File Manager program, FMGR, is used to do this type of processing. If it is, you should wait to do the exercises and examples which follow and come back after the batch processing is finished.

If your system is using MTM, the FMGR program name will also be different. Ask the system manager what the name of the FMGR program for your terminal is. If you cannot get this information, wait until you can use the system console for these exercises. You will be told in Appendix B how to make a copy of FMGR for your terminal but you must learn more commands first.

Earlier, the RUN command was used to schedule FMGR:

*RUN,FMGR	The first FMGR parameter, the <i>input</i> device, defaults to LU1, the system console, or to the LU number of your terminal if you are using MTM. It is not necessary to type it here.
:	

As soon as the colon is printed, you can enter any File Manager command.

FMGR Error Messages

The first command you may want to master is `??`. These two question marks tell FMGR that you want more information about an error message. Suppose the following error appears on the console:

<code>:XX</code>	Type XX in response to the FMGR prompt. FMGR prints error, the
<code>FMGR 010</code>	command you typed, and a ?.
<code>XX?</code>	It then prompts for a command. You type two question marks and press
<code>:??</code>	RETURN.
<code>FMGR 010 INPUT ERROR</code>	FMGR prints the meaning of error 10. XX is not a FMGR command
	which is why the error was printed.

If you type `??` a second time or when there has been no error, it prints `FMGR 000 BREAK`.

If you want to know more about a specific error that is not the most recent error diagnosed, you can type two question marks, a comma, and the error message number. The expansion of the requested error is printed:

```
:??,23
FMGR 023 DUPLICATE PROGRAM NAME.
```

If you want a list of error messages, you type:

```
:??,99
```

FMGR will list all possible errors with additional information about each. It list the errors on the standard device in this case.

If you forget to type the comma following the question marks, FMGR will find the first two characters and ignore the rest until it encounters a comma or RETURN. For example, if you type:

```
:??99
```

FMGR will print the expansion of the last error message it issued if there is one and ignore the 99.

```
:??99,99
```

This will be interpreted as `:??,99`.

```
:??NOTHING
```

This will be interpreted as `??`.

Now that you know how to rescue yourself if you get into trouble, you are ready to learn about creating and storing information on a file.

File Names

When you create a file, you must give it a name that complies with the following rules:

- A name may contain from 1 to 6 characters.
- The first character may not be a number.
- Leading and trailing blanks are ignored. Embedded blanks are not allowed.

- Any printable ASCII character except the plus (+), hyphen or minus (-), comma (,), and colon (:) may be used.
- You must not select a name that is already assigned to a file on the File Manager cartridge you are using.

A File Manager cartridge is a logical entity corresponding to a block of tracks addressed with a unique LU number. The first and last tracks are defined with FMGR commands. The cartridges on your system should already be defined by the system manager. Each cartridge has a File Directory. If you want to find out what files have already been created, you may list the directories on the standard list device by using the DL (directory list) command. The simplest form of this command is:

:DL The system will list names of all files on all mounted cartridges and some information about each file. It will also list the cartridge label and data about the use of tracks and sectors on the cartridge.

Example:

<i>Cartridge</i>						
<i>Reference</i>	<i>Next Available</i>		<i>No. of Sectors</i>		<i>No. of</i>	
<i>Number</i>	<i>Track</i>		<i>Per Track</i>		<i>Directory Tracks</i>	
↓	↓		↓		↓	
<i>Cartridge</i>	<i>Next Available</i>		<i>Last Track</i>		<i>Directory Tracks</i>	
<i>label</i>	<i>Sector</i>		<i>Available</i>			
CR=00013	NXTR=0032	NXSEC=004	#SEC/TR=096	LAST TR= 0202	#DR TR=02	
ILAB=SP00L						

NAME	TYPE	#BLKS/LU	OPEN	TO
JOBFIL	00002	00018	GASP	FMGR
SPLCON	00002	00023	GASP	
SPQL01	00003	00024		
SPQL02	00003	00024		
SPQL03	00003	00024		
SPQL04	00003	00024		
SPQL05	00003	00024		

↑
The file named JOBFIL is open to the GASP and FMGR programs.

↑	↑	↑
<i>File Name</i>	<i>File Type</i>	<i>No. of blocks or an LU No. if type =0</i>

You may list only the files on a particular cartridge by typing the logical unit number of the disc cartridge preceded by a minus sign.

:DL , -2 The FMGR will list information about the files on the cartridge associated with LU 2, the system disc.

Remember, the name you select for a file must be unique for the cartridge you are using. When you read the Batch-Spool Monitor Reference Manual, you will learn how to specify a cartridge label or LU number with your file name. This is a good practice since someone may use the same file name you

have on a different cartridge. If the LU or label is not provided, the File Manager searches all mounted cartridges in the same order they appear in the Cartridge Directory which is kept on the system disc and picks the first file with a matching name. If it finds your name on a cartridge preceding the one your file is on, it will use that file. For now, if you use names which include your name or initials you should not have a problem with duplication.

The Cartridge Directory is described in the Batch-Spool Monitor Reference Manual and a familiarity with it is not required to complete the tasks described in this manual. You can determine the cartridge order by using the CL (cartridge directory list) command to list this directory. It lists the logical unit number, last track assigned to the cartridge, and cartridge reference number of each mounted (active) cartridge. It also gives the name of a program which has locked a cartridge. Here is an example:

:CL Type the command and the cartridge list is printed on the standard list device (the line printer in these examples).

<i>Last Assigned</i>		<i>Cartridge</i>	
<i>Logical Unit</i>	<i>Track</i>	<i>Reference</i>	
<i>Number</i>		<i>Number</i>	
↓	↓	↓	
LU	LAST TRACK	CR	LOCK
62	0202	00062	
02	0175	00002	
13	0202	00019	
15	0202	00100	

← No cartridges are currently locked

Later in this section you will be shown how to list information on the system console; normally cartridge information is listed on the standard list device.

If you are using a system which has more than one cartridge mounted and you do not specify a cartridge when you create a file, the file will be created on the first available cartridge; that is, the first cartridge in the directory which has space for your file.

Create A File

You may use files to store data, lists of FMGR commands, or programs. In this exercise you will use a file to store a few simple sentences on the system disc. Later you will use the same type of file to create and store a FORTRAN program.

The STORE command in FMGR creates a new file and transfers information from another file or logical unit to the new file. The command also may be used to store information or transmit information to a non-disc I/O device such as a line printer or paper tape punch.

The STORE command has two required parameters and four optional parameters. The required parameters define the source and destination of the data to be stored. The optional parameters define the data format. The source may be an input device referred to by its logical unit number or a disc file

referred to by its name. The destination may be an output device or the name of a new disc file which the File Manager will create for you. For example:

:STORE,OLD,NEW A new disc file named NEW is created and the content of the disc file named
:
 OLD is copied to it. The OLD file is unchanged, thus both files contain the
 same information.

In order to try some of the other commands, you will need a file of your own. One way to create such a file is to store data entered at the system console into a disc file. For example:

:ST,1,MINE
THIS IS MY VERY OWN FILE. I CAN TYPE
ANYTHING I WANT AT THE SYSTEM CONSOLE
AND IT WILL BE STORED IN MY FILE.

D^c

This command indicates you are going to create a file named MINE and enter data into the file from the system console which is LU 1. Each line is a record. When you have finished entering the data, you press the control key and the D key simultaneously to tell FMGR the input is complete and to write an EOF mark after the data. There is no prompt character preceding the lines.

Your data is now in the file named MINE. If you doubt it, you may try another FMGR command and list the content of MINE. The command is LI and it has one required parameter, the file name, and one optional parameter to specify the format. If you type D^c immediately after the store command without entering any data at all, the file is not created.

List the Content of a File

:LIST,MINE The content of your file will be listed on the standard list device, LU 6. The
:
 optional parameter has a default value which causes the file content to be
 listed in ASCII because the data you typed is ASCII. The explanation of
 exactly how this occurs is much more complicated but you can read about that
 in the reference manual when you are ready.

Go to the standard list device (in the example it is a line printer) and look at the listing. Each line is a record and lines are numbered starting with 1.

MINE T=00003 IS ON CR00002 USING 00001 BLKS R=0000
0001 THIS IS MY VERY OWN FILE.
0002 I CAN TYPE ANYTHING I WANT AT
0003 THE SYSTEM CONSOLE AND IT WILL
0004 BE STORED IN MY FILE.

List Information on the System Console

If you want to list file information on the system console, you can use command LL to change the standard list device from the line printer to the system console. Like this:

```
:LL,1      Type LL followed by the LU number of the device to which you want standard
:          listings to go. In this example it is LU 1, the system console.
```

If you want to change the list device back to the standard, you simply type:

```
:LL,6      The standard list device is now the line printer, LU 6, again.
```

You can only change the device when FMGR has prompted you for a command.

A list device must be capable of receiving output, but if you change the list device to LU 2, the system disc, or LU 5, the standard input device, no error message is printed until you give a command that uses the standard list device. Then an error will be printed and the File Manager is aborted.

If you are using a MTM terminal, you must use your terminal LU number instead of LU 1 to get the listing on your terminal.

The Editor

When you examine the printed listing of your file suppose you discover that you made some errors or that you want to add something. Don't despair. . .you will not have to type the whole file again. You can use another program, EDITR, the Interactive Editor. EDITR runs in background mode like FMGR and you may schedule the EDITR program with the RTE command, RUN, or, if you are currently using the File Manager, you may use the File Manager RUN command. If you are using a MTM terminal, you should ask the system manager for the editor program name to be used at your terminal. If there is not one, wait until you can use the system console to run EDITR.

RUN is followed by the name of the program you want to run (in this case, EDITR) and a list of parameters which vary according to the program. The EDITR program has two optional parameters, the logical unit number of the command input device and the maximum number of characters per line. The default values for these parameters are the logical unit of your terminal, and a line length of 150 characters. When running the EDITR from File Manager the first parameter defaults to the first parameter of the RUN command you used to schedule the FMGR program.

```
:RU,EDITR          The EDITR first requests the name of the source file to be edited.
SOURCE FILE?
/
```

EDITR has a different prompt character than the File Manager. It is the slash, /. When EDITR is ready for a request, it will print the slash prompt character. One way to determine which program you are in and when you are communicating with the system, is to recognize the different prompt characters.

*	RTE system prompt
:	FMGR prompt
/	EDITR prompt

When designing your own application programs which require conversational input, it is advisable to use yet another prompt character for clarity.

EDITR allows you to edit files in several ways; however, in this manual only the line-by-line edits are discussed. Later you may consult the RTE-III Interactive Editor Reference Manual to learn about the character and pattern editing commands.

The term *pending line* is used by EDITR to refer to the particular line in the file which is currently being edited. EDITR automatically maintains the number of the line which is the pending (current) line. You can use EDITR commands to make any line in your file the pending line.

If you type the name of the file you just created, EDITR displays the first line of the source file, the initial pending line.

```
:RU,EDITR          Type the name of your file. Press RETURN. EDITR prints the first line and
SOURCE FILE?       prompts you for a command.
/MINE
  THIS IS MY VERY OWN FILE.
/
```

EDITR has two work areas on the disc which it uses for editing files. These are described in the RTE-III Interactive Editor Reference Manual. Since EDITR uses the work areas, your original file is unchanged unless, when you are finished editing, you tell the Editor to replace that file with your edited version. You may prefer to move the new version into a new or different file and save the previous version. There are other ways to use the new version and they will be described in Section III.

EDITR commands consist of one character, usually the first letter of the command name. Suppose you want to replace the pending (current) line. You can do this with the R (replace) command:

```
/RTHIS IS LINE 1 OF MY VERY OWN FILE
/P
```

Type R and then type the new line just as you

```
THIS IS LINE 1 OF MY VERY OWN FILE
```

want it to appear in the file. Then type the command P (pending line) to display the new pending line. EDITR prints the line.

If you want to list all the lines in the current version of the file, you use the L (list) command. You specify how many lines you want to list in addition to the pending line.

```
/L2
THIS IS LINE 1 OF MY VERY OWN FILE I CAN
TYPE ANYTHING I WANT AT THE SYSTEM
CONSOLE AND IT WILL BE STORED IN MY FILE
```

The EDITR lists the pending line and two lines which follow it. Line 3 is now the pending line.

```
/
/L20
THIS IS LINE 1 OF MY VERY OWN FILE I CAN
TYPE ANYTHING I WANT AT THE SYSTEM
CONSOLE AND IT WILL BE STORED IN MY FILE
EOF
```

OR

If the number of lines you specify is greater than the number of lines in the file, all lines in the file and an EOF indicator will be listed.

```
/
/L0
THIS IS LINE 1 OF MY VERY OWN FILE
I CAN TYPE ANYTHING I WANT AT
```

OR

If you type 0 for the number of lines, EDITR interprets 0 as a 1.

Suppose you want to change the third line in the file but you do not want to change the second line. You can make line 3 the pending line by simply typing the line number.

```
/3
THE SYSTEM CONSOLE AND IT WILL
/
```

EDITR makes line 3 the pending line and displays it.

You could also have made line 3 the pending line by using the + or / command. For example:

```
/+2
THE SYSTEM CONSOLE AND IT WILL
/
```

EDITR skipped the pending line and the one following it and made the next line the pending line.

```
//2
THE SYSTEM CONSOLE AND IT WILL
/
```

This is exactly the same as the previous command. It is provided for your convenience since the + character requires the shift key and the slash does not.

If you try to set the pending line pointer to a number greater than the number of lines in the file, the pointer will be set at the EOF and there will be no pending line.

```
/10
EOF
/P
?? ← Error Message
/2
  I CAN TYPE ANYTHING I WANT AT
/P
  I CAN TYPE ANYTHING I WANT AT
```

Since there is no pending line, the Editor prints an error message if you try to print it. You may set the pending line to 2 and then display it back.

Similarly, if you skip more lines than there are in the file following the pending line you will be in this same predicament.

```
/2
  I CAN TYPE ANYTHING I WANT AT
/+6
EOF
/P
??
```

What are other ways you may alter the file? Two very useful commands allow you to insert a line. The I command inserts a line immediately preceding the pending line, and the Δ (space) command inserts a line immediately following the pending line. Let's try them. Suppose the pending line is 2.

```
/2
  I CAN TYPE ANYTHING I WANT AT
/ΔMY LEISURE ON ← insert line after
/P                               pending line
  MY LEISURE ON
/2
  I CAN TYPE ANYTHING I WANT AT
/L4
  I CAN TYPE ANYTHING I WANT AT MY LEISURE
  ON THE SYSTEM CONSOLE AND IT WILL BE
  STORED IN MY FILE
EOF
/2
  I CAN TYPE ANYTHING I WANT AT
/II'M SO CLEVER
/1
  THIS IS LINE 1 OF MY VERY OWN FILE
/L6
  THIS IS LINE 1 OF MY VERY OWN FILE
  I'M SO CLEVER
  I CAN TYPE ANYTHING I WANT AT
  MY LEISURE ON
  THE SYSTEM CONSOLE AND IT WILL
  BE STORED IN MY FILE
```

You type a space and then the line which you want to insert after the current line. Type P and you will see that the pending line is the line you inserted.

If you make line 2 the pending line and list four lines, the new line is there.

Make line 2 the pending line again. You may insert a line before the pending line by typing I and then the line you want to insert. Now make line 1 the pending line and list six lines. The file looks like this.

The I and Δ commands insert one line only, but you can continue to insert lines by repeating these commands.

```

/6
    BE STORED IN MY FILE
/ΔTHE CONTENT
/ΔOF THIS FILE
/ΔIS NONSENSE.
/1
    THIS IS LINE 1 OF MY VERY OWN FILE
/L9
    THIS IS LINE 1 OF MY VERY OWN FILE
    I'M SO CLEVER
    I CAN TYPE ANYTHING I WANT AT
    MY LEISURE ON
    THE SYSTEM CONSOLE AND IT WILL
    BE STORED IN MY FILE
    THE CONTENT
    OF THIS FILE
    IS NONSENSE.
/

```

Make line 6 the pending line. Add three lines by inserting a line after the pending line. Each time a line is added, it becomes the new pending line. Go back to line 1 and list all the lines.

You may delete lines using the minus (−) command. If you specify a number after the minus, EDITR deletes that many lines beginning with the pending line. The line following the last deleted line becomes the pending line.

```

/1
    THIS IS LINE 1 OF MY VERY OWN FILE
/-
    I'M SO CLEVER
/-5
    THE CONTENT
/

```

Make the pending line 1.

Delete the pending line. This is the same as requesting −0 or −1. The new pending line is displayed. As you are probably bored with the remaining lines of your file, having read them so often, you may type −5 to delete the next 5 lines, including the pending line.

The lines of your file are not renumbered until you reference a line preceding the current pending line. For example, each time you reset the pending line to 1 in the examples above, the lines of the file were renumbered. After deleting five lines from the file you may use the N (number) command to find out the pending line number.

```

/N
    7
/1
    THE CONTENT
/N
    1
/L9
    THE CONTENT
    OF THIS FILE
    IS NONSENSE.
EOF
/

```

The pending line number is still 7 since you have not referenced a line preceding the pending line. However, if you reset the pointer to the beginning of the file and make line 1 the pending line, all lines are renumbered and now the same line is line 1.

See the RTE-III Interactive Editor Reference Manual to learn more about the renumbering process.

After you finish editing the file, you must tell the Editor to save the new version or all your work will be lost when you return to the File Manager. This is easy to remember because the only way to exit EDITR without saving the new version is by using the A (abort) command. Each of the normal exit commands begin with an E and the letters which follow the E define where the new version is to be moved. At this time only the EC and ER commands will be discussed.

If you want to store the new version in the file that contained the original version, use the ER (end with replace) command.

```
/ER          The new data replaces the original data in MINE and control is returned to the
END OF EDIT  File Manager which prompts for a new command.
:
```

If you want to create a new file to contain the new version, you use the EC (end with create) command. You must supply a file name in the same way and obey the same rules for file names described with the File Manager STORE command.

```
/ECNEWFIL    A file named NEWFIL will be created and the new data will be stored in it
END OF EDIT
:
```

If you scheduled EDITR from RTE, no prompt is printed after END OF EDIT until you hit a key to get the system's attention. Then an asterisk is printed.

You may exit EDITR without saving the new version by typing the A command. This command aborts the Editor and, because you scheduled EDITR with the FMGR RUN command, returns you to File Manager. If you want to abort the Editor immediately after the SOURCE FILE? prompt, you must type a space and carriage return. The computer will then print EOF or a line of information and a prompt. Type A in response to the first / prompt and the message EDITR ABORTED will be printed. You are now returned to the File Manager or RTE.

More FMGR Commands

There are five more File Manager commands which you may need while doing the examples in this manual. These are PURGE, RNAME, RTRACKS, DUMP, and EXIT.

The PURGE command removes a file from the system. Obviously it is wise to be careful when using this command. You would not want to purge a file accidentally and lose the data.

:PURGE,MINE The file named MINE no longer exists in the File Manager directory.
:

If the message FMGR -006 is typed, the system could not find a file with the name you typed. Check the name and try the command again. If the message FMGR -007, is typed, it indicates that a file security code is needed and you have not provided it. Unless you know the security code you cannot purge the file. Security codes are not discussed in this manual but are described fully in the Batch-Spool Monitor Reference Manual. This error will probably not occur unless you accidentally type the name of an existing file rather than one of your own.

If you want to change the name of a file but do not want to alter the file content in any way, you may use the RNAME command. You simply give the current name of the file followed by the new name you would like to use.

:RNAME,NEWFIL,FRED The file previously named NEWFIL will now be named FRED and you will no longer have a file named NEWFIL.

You may get FMGR -006 and FMGR -007 errors if you do not type the current name of your file correctly. You may get error FMGR -002 if the new name you give your file has already been used for another file.

You may not rename a file if the file is open, that is, if it is being used by a program. If you do you will get the error FMGR -008. Later you will want to know more about the subparameters, security code and cartridge reference, associated with this command. You may read about this in the Batch-Spool Monitor Reference Manual.

The RTRACKS command may be used as a FMGR command and also as a RTE command. The purpose of the command is to release disc tracks assigned to programs. It is a good idea to use this command in relation to EDITR after you have finished using the program. This habit will insure that the disc work area will not get filled up with bits and pieces of unused data or other information.

:RT,EDITR Type the command and the program name separated by a comma.
OR

***RT,EDITR** RT is also an RTE command.

When you are finished doing all the file management tasks you want to do, you tell the File Manager you want to exit with the EXIT command.

:EXIT This message is printed indicating you are no longer running the FMGR
\$END FMGR program.

In order to do something else with RTE, you must hit any key on the console to get the RTE prompt character,*.

The DUMP command may be used in several ways. You can:

- dump from a logical unit to another logical unit or a file,
- dump from a file to another file or a logical unit.

For example:

<code>:DUMP,5,6</code>	This command will read from LU 5, the standard input device, and dump the content of the tape on LU 6, the standard list device.
<code>:DUMP,FRED,1</code>	You may list the content of the file named FRED on the system console, LU1.
<code>:DUMP,FRED,4</code>	You may want to dump the content of a file to the standard output device, a paper tape punch or magnetic tape device.

The DUMP command differs from the STORE command in that it does not create a file. If you DUMP from one file to another, both files must exist. You do not need to create files except by using the STORE command in order to do the tasks described in this manual. The Batch-Spool Monitor explains file creation completely and you can consult it when you are ready to create files in other ways.

Program Development

At this point, a general overview of the process of program development with the RTE system might be helpful. Since you have written programs before, you probably know about the source files, relocatable binary code (sometimes called the object code), and memory image code. But in case you do not, here is a quick review. Do not be concerned if you do not understand all of the information completely now. It will be discussed again in detail as you do the tasks in the next section.

There are several types of files managed by the File Manager; however, you will only need four types (3, 4, 5, and 6) to create, compile or assemble, and run your program.

Program source files contain the original source statements written in FORTRAN, ALGOL, or Assembly Language. The first step in developing your program will be to transmit the source statements to the computer. There are many ways to do this but initially this manual shows you how to put these statements into a File Manager ASCII data (type 3 or 4) file. BASIC source programs are also stored in files but are processed differently from the other languages.

After the source program is created and the content has been checked and edited to your satisfaction, you will compile or assemble the program. A program may be composed of a main program and segments. The main program and each segment may also have subroutines appended to them. For simplicity, the initial discussion will be about a main program without any segments.

Before you can run the program, you must execute another program, LOADR, which takes the relocatable binary code, appends all required subroutines either supplied by you or from the system library, and creates one more version of the code. This version, the RTE memory image code, is used when you actually run the program. An ID segment, the next topic discussed in this section, is also created by LOADR.

You may save the memory image code in a type 6 file by using another FMGR command, SP. This command is explained in the next section.

Thus, when you have finished the program development in the following pages, three forms of the program will be stored in the disc files: the source program, the compiled relocatable binary code, and the memory image code containing all appended subroutines.

ID Segments

Before you can run a program you must make that program known to the system. The system saves information about the memory image form of the program in area called the ID segment. The number of ID segments is defined at system generation time.

When you schedule a program from the system with RUN or ON, the system searches the ID segments until it finds a program with the name you provide. If it does not find one it types the message NO SUCH PROGRAM. In other words, ID segments serve as a directory to programs in the system.

Programs may be permanent or temporary. The ID segments of permanent programs are saved on the disc as part of the system and each time the system is loaded the ID segments are restored to memory. The ID segments of temporary programs are only saved in memory and when the system is reloaded they disappear.

The system creates a program's ID segment during the system generation or when you use the Loader program (LOADR) on-line or the File Manager RP command after the system is loaded.

LOADR links relocatable object code and required subroutines to create memory image code, saves it on the disc, and puts information into the ID segment which tells where the program is located, its priority, name, and other information. Dynamic program status information about the program is also kept in the ID segment.

The RP command locates an unused ID segment and enters information in it which points to a program stored in a type 6 file. All programs entered in ID segments with the RP command are temporary programs.

In the next section you will use the Loader to create an ID segment for a temporary program. Once the program has an ID segment you may run it by simply using the RTE RUN or ON command or the File Manager RUN command.

If you want to delete a temporary program from the system, you use the RTE command, OFF, with the program name and specify 8 as the second parameter. For example:

***OFF ,PROGX ,8** The temporary program named PROGX is deleted and the ID segment is now free to be used by another program.

If you are running FMGR, you just type OFF and the program name. For example:

: OFF ,PROGX This command in FMGR is the same as the system *OFF,PROGX,8 command.
PROGX ABORTED
:

PROGRAM PREPARATION

SECTION

II

If you do each step in this section, you will create a program source file, compile the program, load, and run it.

Try doing the examples shown in the boxes. You should have a FORTRAN IV Reference Manual handy to use if you have questions about statement formats or errors which result from compilation. This manual tells you only about FMGR, EDITR, and RTE errors which you may encounter. If something happens which is not described with the example, turn to Section IV.

Remember:

- Press any key to get the RTE prompt character.
- Terminate each line of your source program by pressing RETURN.
- Terminate the Editor with an EC or ER command.
- FORTRAN statements must be entered in the format shown in the examples, the statement numbers in columns 1 to 5, a blank in column 6, and the statement in columns 7 through 72.

Names such as SOURC1, SOURC2, and RELOC are used in the examples to emphasize the form or purpose of the named program or file. When practicing the commands and techniques described in this manual, you should use names which you can remember easily and that are not likely to duplicate other names in the system. You can use your initials or some letters from your name and combine them with a letter which identifies the purpose of the name. For example, if your initials are XYZ you can name your program XYZP and your source file XYZS.

At this time, if you are using a MTM terminal, you may want to skim the discussion of MTM in Section III so you can anticipate some differences you will encounter between your terminal and the examples.

STEP 1 . . . Create a Source File

You can create a source program file in one of several ways:

- punch the program on cards off-line and then store the card content on a disc file using the STORE command,
- type the program to a mini-cartridge off-line and then store the tape content on a disc file using the STORE command,
- type the source statements at your terminal and store them in a disc file using the STORE command,
- type the source statements at your terminal under control of the EDITR program and create a disc file containing the statements at the end of your Editor session.

The last method is the most efficient because you can edit your mistakes (if you make them) while you are creating the file. This the method demonstrated in the example which follows.

If you reply with a space (Δ) and RETURN to the SOURCE FILE? prompt, EDITR moves the contents of the current LS area to its work area. The logical source (LS) area is an area consisting of one or more tracks for use in program development. The system maintains a pointer to one currently active logical source area. The pointer may also be cleared to indicate that there is no active logical source area.

If you want to create a new file using the Editor, you must clear the LS pointer before scheduling EDITR so EDITR will initialize a clear work area rather than copying the content of the current logical source tracks to its work area. The Editor prints EOF instead of a pending line to indicate the area is empty.

It is most convenient to do all operations from File Manager so first use the RTE command, RUN, to run the FMGR program. Then use the LS command as shown in Figure 2-1 to clear the LS pointer.

Next use the File Manager RUN command to run EDITR. When EDITR asks for the name of a source file, type a space and press RETURN. If EDITR prints anything other than EOF, you have forgotten to clear the LS pointer. You can still clear the EDITR work area by typing -32000 (delete 32000 lines). It then prints EOF and you can proceed.

To enter new source statements, type a space, the statement (beginning in column 1), and then press RETURN. EDITR prompts for the next command. When you finish, list your program. If it is correct, use the EC command to create a File Manager file named SOURC1 and to exit the EDITR program.

You can use the File Manager LIST command to list the program to the standard list device. Figure 2-2 shows the line printer listing. The Editor will also list to the printer if you specify a LU number following the L command. For example, /L10,6 will print 10 lines to LU6, the standard list device.

FORTTRAN IV requires a control statement preceding the program. In the example, only the L parameter is used. This parameter indicates you want a listing of the program. See Table 3-2 in Section III for a list and description of the other parameters.

The first line of code should contain the name of your program. In FORTRAN it is specified in the PROGRAM statement.

The error messages which you may encounter are summarized in Table 2-1. The meaning is given in terms of the particular situation shown in the example. If you take the recommended action and are still having trouble, you may have to consult the system manager.

*RU,FMGR	◀ <i>Run the File Manager.</i>
:LS	◀ <i>Clear the logical source pointer.</i>
:RU,EDITR	◀ <i>Run the Editor.</i>
SOURCE FILE?	
/Δ	◀ <i>Type a space and press RETURN.</i>
EOF	◀ <i>EDITR indicates the work area is clear by printing EOF.</i>
/ΔFTN4,L	◀ <i>Type a space and the FORTRAN IV control statement.</i>
/Δ PROGRAM PROG1	◀ <i>Program name is PROG1.</i>
/Δ DIMENSION NAME(10)	◀ <i>For each line of your program, type a space and the line, then press return.</i>
/Δ DO 20 I=1,10	
/Δ NAME(I)=2HΔΔ	
/Δ 20 CONTINUE	
/Δ WRITE (1,30)	
/Δ 30 FORMAT ("PLEASE TYPE YOUR NAME AND PRESS RETURN.")	
/Δ READ(1,40) NAME	
/Δ 40 FORMAT (10A2)	
/Δ WRITE (1,50) (NAME(I),I=1,10)	
/Δ 50 FORMAT ("YOUR PROGRAM WORKS, ", 10A2)	
/Δ END	
/Δ END\$	◀ <i>Indicate end of source program.</i>
/1	◀ <i>Set the pending line to 1.</i>
FTN4,L	
/L14	◀ <i>List the pending line plus fourteen more.</i>
FTN4,L	
PROGRAM PROG1	
DIMENSION NAME (10)	
DO 20 I=1,10	
NAME(I)=2H	
20 CONTINUE	
WRITE (1,30)	
30 FORMAT ("PLEASE TYPE YOUR NAME AND PRESS RETURN.")	
READ(1,40) NAME	
40 FORMAT (10A2)	
WRITE (1,50) (NAME(I),I=1,10)	
50 FORMAT ("YOUR PROGRAM WORKS, ", 10A2)	
END	
END\$	
EOF	◀ <i>EDITR indicates the end of file.</i>
/ECSOURC1	◀ <i>Create a file named SOURC1 which contains the source program and exit EDITR.</i>
END OF EDIT	◀ <i>List the content of SOURC1 on the standard list device.</i>
:LI,SOURC1	
:	

Figure 2-1. Example of Creating a Source File

Table 2-1. Error Messages, Step 1

<u>ERROR</u>	<u>MEANING</u>	<u>ACTION</u>
NO SUCH PROGRAM	The program you requested does not exist.	Check the name and try the command again.
ILLEGAL STATUS	FMGR is already running.	Determine whether anyone else is running FMGR. If so, try later. If you were running it previously, there may be a device down. Check the line printer and try *UP,6.
??	You have typed an invalid EDITR command.	Look at the example, determine what the correct command is, and type it again.
FILE MANAGER-002	If your system already has a SOURC1 file or if you already created one, you will get this message.	Use another file name in the task examples.
FILE MANAGER-006	If you type a name instead of a space in response to the EDITR prompt for a source file, you may get this message. Or, there is not room on the disc to create SOURC1.	Type a space and press RETURN when the prompt is repeated. See the system manager.
FILE MANAGER-015	If you type a new source file name which does not obey the syntax rules described in Section I, you will get this error. (i.e., /EC12XX)	Try the command again with a valid name.
FMGR-017	You probably forgot to type the file name after the LI command.	Try typing the command again.
FMGR-006	This error occurs if you use LI and type the file name incorrectly.	Check your input for accuracy. Correct and type the command again.
FMGR 010	You have probably typed a syntactically incorrect or unrecognized FMGR command.	Check the command syntax and try again.
FMGR 056	A FMGR command parameter is not valid.	Check the parameters and try the command again.

Table 2-1. Error Messages, Step 1 (Continued)

<u>ERROR</u>	<u>MEANING</u>	<u>ACTION</u>
IO07 FMGR nnnnn FMGR ABORTED	You have tried to do an I/O operation to an inappropriate device—for example, storing file data to the input device, LU 5.	FMGR has been aborted so you must type *RU,FMGR. Find your mistake and try the command again.

```

SOURC1 T=00004 IS ON CR00002 USING 00002 BLKS R=0014

0001  FTN4,L
0002      PROGRAM PROG1
0003      DIMENSION NAME (10)
0004      DO 20 I=1,10
0005          NAME(I)=2H
0006      20 CONTINUE
0007      WRITE (1,30)
0008      30 FORMAT ("PLEASE TYPE YOUR NAME AND PRESS RETURN.")
0009      READ(1,40) NAME
0010      40 FORMAT (10A2)
0011      WRITE (1,50) (NAME(I),I=1,10)
0012      50 FORMAT ("YOUR PROGRAM WORKS, ", 10A2)
0013      END
0014      END$

```

Figure 2-2. FORTRAN Program Listing

STEP 2 . . . Edit a Source File

Examine the listing of your source file. Are there any errors? If so, you can use EDITR to correct them before you compile. One line is replaced in the example in figure 2-3. You may want to do the exercise even if you don't need to change your file.

<pre> :RU,EDITR SOURCE FILE? /SOURC1 FTN4,L /L14 FTN4,L PROGRAM PRG1 DIMENSION NAME (10) DO 20 I=1,10 NAME(I)=2H 20 CONTINUE WRITE (1,30) 30 FORMAT ("PLEASE TYPE YOUR NAME AND PRESS RETURN.") READ(1,40) NAME 40 FORMAT (10A2) WRITE (1,50) (NAME(I),I=1,10) 50 FORMAT ("YOUR PROGRAM WORKS, ", 10A2) END END\$ EOF /12 50 FORMAT ("YOUR PROGRAM WORKS, ", 10A2) /R 50 FORMAT("GOOD. YOUR PROGRAM WORKS, ", 10A2) /P 50 FORMAT ("GOOD. YOUR PROGRAM WORKS, ", 10A2) /ECSOURC2 END OF EDIT :LL,1 :LI,SOURC2 SOURC2 T=00004 IS ON CR00002 USING 00002 BLKS R=0014 0001 FTN4,L 0002 PROGRAM PRG1 0003 DIMENSION NAME (10) 0004 DO 20 I=1,10 0005 NAME(I)=2H 0006 20 CONTINUE 0007 WRITE (1,30) 0008 30 FORMAT ("PLEASE TYPE YOUR NAME AND PRESS RETURN.") </pre>	<p>◀ Use FMGR RU command to run EDITR.</p> <p>◀ EDITR requests Source file name.</p> <p>◀ Type SOURC1 and return.</p> <p>◀ EDITR lists first line of file.</p> <p>◀ Type L and 14 to list 14 lines of the file in addition to the pending line.</p> <p>◀ EDITR prints EOF to indicate end of file.</p> <p>◀ Type 12 to make line 12 the pending line.</p> <p>◀ Type R and a new line.</p> <p>◀ Type P to print the new pending line.</p> <p>◀ Type EC to create a new file SOURC2 and exit the Editor program.</p> <p>◀ Type LL,1 to change the list device to LU1</p> <p>◀ List the new file.</p>
--	--

Figure 2-3. Example of Editing a Source File

```

0009      READ(1,40) NAME
0010      40 FORMAT (10A2)
0011      WRITE (1,50) (NAME(I),I=1,10)
0012      50 FORMAT ("GOOD. YOUR PROGRAM WORKS, ", 10A2)
0013      END
0014      END$

```

Figure 2-3. Example of Editing a Source File (Continued)

Now you have two files, SOURC1 containing the original version and SOURC2 containing the new version. If you have a very large program, it is a good idea to save the previous version in case you accidentally purge the new file or make a serious editing error. You can use the old file to recover more easily. You will be using the SOURC2 file in the steps which follow.

After you get the new listing, you should check it again for accuracy. If it is correct, proceed to step 3. If not, edit the file again but this time use the ER command to exit so the latest file version is in SOURC2.

Table 2-2. Error Messages, Step 2

<u>ERROR</u>	<u>MEANING</u>	<u>ACTION</u>
FMGR-006	You will get this error if you do not type the EDITR name correctly.	Check what you typed and try again.
FMGR 010	If you type an invalid FMGR command you will get this error.	Type the command again correctly.
FILE MANAGER ERROR -02	Your system may already have a file named SOURC2.	Retype the EC command. Use a different file name and remember to use that name in the steps that follow this one.
FILE MANAGER ERROR -06	You have typed the name of your source file incorrectly.	Check the name and type it again.
??	You have typed an invalid EDITR command.	Look at the example and determine what the correct command is and type it again.
IO07 FMGR nnnnn FMGR ABORTED	You tried to change the list device to a logical unit number which is not an output device.	In these examples you should only make the list device LU 6 or 1. Since FMGR is aborted, you must schedule it again and then repeat the command.

STEP 3 . . . Compile a Program

Now you are ready to run the FORTRAN IV compiler program, FTN4, from the File Manager with the RUN command. You will need to specify, as parameters in the RUN command, the names of the source, list and relocatable files. Logical unit numbers may be used in place of file names in the RUN command. Below is an illustration of invoking the FTN4 compiler:

input *output*
 ↓ ↓
RU,FTN4,SOURC9,1,RELOC
 ↑
 list

In this example the source file name is SOURC9. The compiler listing is output to the system console's display, and the compiled relocatable file is named RELOC.

After you compile the program, check the listing for compiling errors. If there are any errors, use the EDITR to correct the source statements, purge the relocatable file using the PURge command, and re-compile your program.

In the example below, SOURC2 is specified as the source file, LIST1 is the filename of the compiler listing, and RELOC the file containing the relocatable binary code. Table 2-4 contains some errors which you might possibly encounter during this step. Figure 2-6 is an example of an RTE FORTRAN IV compiler listing.

```
:RU,FTN4,SOURC2,LIST1,RELOC
```

```
/FTN4: $END
```

◀ *Run FTN4 using SOURC2 for the source file, LIST1 for the compiler listing, and RELOC for the relocatable file.*

◀ *The FORTRAN IV compiler prints this message when it completes the compilation.*

Figure 2-4. Compile FORTRAN IV Program

Table 2-3. Error Messages, Step 3

<u>ERROR</u>	<u>MEANING</u>	<u>ACTION</u>
FMGR-006	You typed the program name FTN4 incorrectly. File Manager cannot find a program with the name you typed so it looked for a file of that name and did not find one.	Type the RUN command again.
FMGR-002	A file name RELOC already exists on your system.	Use some other file name and try the SA command again.
FMGR 010	You have typed an invalid FMGR command.	Try again with a valid command.
FMGR 050	You forgot to type one of the parameters of the SA command.	Check to see that you have the LG parameter and the file name. Type the command again.
FMGR 056	You will get this error if you do not include the program name in the RUN command or if you type the first parameter of the SA command incorrectly.	Check the command, correct your mistake when you type it again.
I/O ERR TO EQT #5	If you do not provide the 2 parameter, (LU 2 = input device) File Manager will try to read the source program from LU 5, the paper tape reader. You will get this error if there is not a paper tape mounted and ready.	Type the command again with both the 2 and 99.
IO06 FTN4 nnnnn ABEND FTN4 ABORTED	You forgot to define the LG area with the LG command in Step 3.	Define the area now and type the RUN command again.
ERROR 05	This error will be printed on the standard list device but not at your terminal. You forgot to declare the LS area.	Go back to step 3 and type the MS command. Then type the RUN command again.

PAGE 0001

FTN4 COMPILER: HP24177 (SEP. 1974)

```
0001 FTN4,L
0002     PROGRAM PROG1
0003     DIMENSION NAME (10)
0004     DO 20 I=1,10
0005         NAME(I)=2H
0006     20 CONTINUE
0007     WRITE (1,30)
0008     30 FORMAT ("PLEASE TYPE YOUR NAME AND PRESS RETURN.")
0009     READ(1,40) NAME
0010     40 FORMAT (10A2)
0011     WRITE (1,50) (NAME(I),I=1,10)
0012     50 FORMAT ("GOOD. YOUR PROGRAM WORKS, ", 10A2)
0013     END
```

*** NO ERRORS** PROGRAM = 00122 COMMON = 00000

No. of compilation errors

PAGE 0002 PROG1 FTN4 COMPILER: HP24177 (SEPT. 1974)

0014 END\$

Figure 2-5. FORTRAN Listing

STEP 4 . . . Load and Run a Program

The RTE Loader, LOADR, links your programs with system library subroutines and converts them to memory image code. It then stores this version of the code on the disc and creates an ID segment so it can later be loaded into a memory partition and run.

LOADR always takes its input from a special area of the disc called the LG area. You may specify the size of the LG area by the FMGR command :LG,n where n is the number of tracks you need for your program. Generally, an area of 1 or 2 tracks is large enough for a moderate to large main program.

Once the LG area is set up, you must move your relocatable file to it. This is done by the MR FMGR command. For example, to move a relocatable file named REL9 to the LG area of the disc, type:

```
:MR,REL9
```

In the example in figure 2-6, LOADR is run from File Manager. Only the input parameter is specified. It is 99 which indicates that input is from the LG area. By default your program is a background temporary program using only local common or no common area. It is loaded into any available partition.

If the Loader completes the relocation successfully, it prints a message and then you can run your program. When the Loader terminates, the LG pointer is reset to the beginning. The LG area still exists and is ready to use again.

:LG,2	◀ Declare LG area with size of 2 tracks.
:MR,RELOC	◀ Move RELOC file to LG area of disc.
:RU,LOADR,99	◀ Type the FMGR RU command to run LOADR.
/LOADR:PROG1 READY	◀ When LOADR completes successfully it prints this message.
/LOADR:\$END	
:RU,PROG1	◀ Type RU and your program name.
PLEASE TYPE YOUR NAME AND PRESS RETURN.	◀ Your program asks you to type your name.
KIM	◀ You do.
GOOD, YOUR PROGRAM WORKS, KIM	◀ The program prints a message to you.
:SP,PROG1	◀ Save the memory image program.
:EX	
\$END FMGR	
:EX	◀ Type EX to exit File Manager.
\$END FMGR	◀ FMGR is terminated.

Figure 2-6. Loading and Running a Program

If your program has a logic error, you can go to Step 6 which describes the way to fix it. If not, you can use the FMGR command, SP, to save your memory image program in a File Manager file. The name of the file is the same as the first 5 characters of the program name. If you save your program with this command you can run it the next time the system is loaded without recompiling your source code. If you do not save it, the program disappears when the system is loaded because it is a temporary background program.

If the LOADR program is already being used by someone else, File Manager will suspend until the Loader has terminated and then schedule it for you. See Section IV, Symptoms, for more information.

Table 2-4. Error Messages, Step 4

<u>ERROR</u>	<u>MEANING</u>	<u>ACTION</u>
/LOADR: L 10 /LOADR: LOADR ABORTED	If you did not type the 99 in the RUN,LOADR command, you will get this error.	Type the command with the 99.
/LOADR: L 08 /LOADR: LOAD	If you did not define the LG area, if the LG area does not have relocatable code in it, or if your program does not have a transfer address in it, you will get this error.	Use the RTE abort command, AB, to terminate the Loader. (*AB) Then repeat steps 3, 4, and 5.
NO BLANK ID SEGMENTS	The program cannot be loaded now because there are no ID segments available.	Consult with your system manager to decide how an ID segment can be made available.
/LOADR: DUPLICATE PROG NAME — PROG1 /LOADR: . . OG1 READY /LOADR:\$END	There is already a PROG1 program in your system so the Loader assigns the name,..OG1 to your program.	If this is the second time you have received this message, the LOADR will abort. You must use EDITR to change the name of your program and repeat steps 3, 4, and 5.
WAITING FOR DISC SPACE	The Loader is waiting until it can allocate a track for its operation. The Loader is suspended.	Wait or see the system manager.
FMGR 050	You did not supply the program name parameter with the SP command.	Type the command again and supply the program name.
FMGR 014	If you type the program name incorrectly, you will get this error because FMGR cannot find a program with the name you typed.	Try again with the correct name.

PROG1 44000 44171

FMTIO 44172 45422

CLRIO 45423 45425

.FLUN 45426 45446

.XFER 45447 45512

DBLE 45513 45547

SNGL 45550 45615

FRMTR 45616 50355

.OPSY 50356 50415

.XCOM 50416 50466

.XPAK 50467 50663

.PACK 50664 50777

.ZRLB 51000 51040

ENTRY POINTS

*EXEC 10162

*\$LIBR 10372

*\$LIBX 11053

*.MPY 100200

*.DLD 104200

*.DST 104400

*.ENTP 26607

*FLOAT 105120

*IFIX 105100

*PROG1 44000

.

.

*.OPSY 50356

*.ZRLB 51000

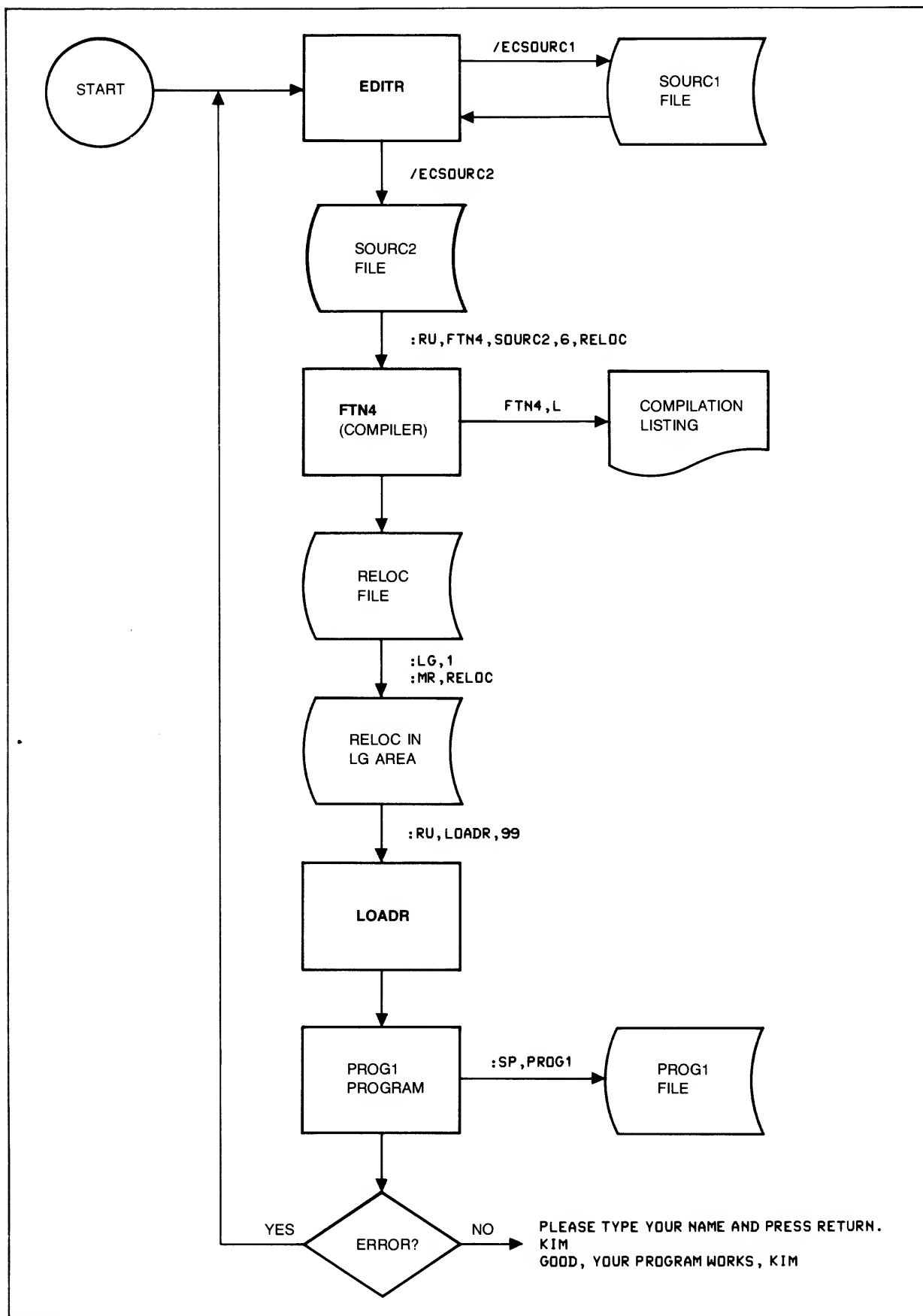
*.DFER 45447

*.XPAK 50500

*.PACK 50664

*.XCOM 50416

Figure 2-7. LOADR Listing



7700-351

Figure 2-9. Developing a Program

STEP 5 . . . Correct Program Logic Errors

In the development of most programs, errors in syntax (spelling, coding rules, proper placement of commas, etc.) and program logic errors are common. Step 2 in this section explains the process of correcting syntax errors by using the Editor. You also use the Editor to correct logic errors but you should do some other things first.

Recall that in preparing the program you compiled the source code version, SOURC2, and stored the relocatable binary version in a file named RELOC. Then you created a memory image version and ID segment using the Loader and saved this version in a file named PROG1.

Assume that you now discover an error in the logic of your program. The sequence of instructions that follows deletes the binary versions of your program from the system:

<code>:PURGE,RELOC</code>	Remove the file named RELOC from the File Manager directory, also remove
<code>:PURGE,PROG1</code>	PROG1.
<code>:OFF,PROG1</code>	Clear the ID segment created for the temporary program PROG1. If you do not
<code>PROG1 ABORTED</code>	do this from FMGR, you must type *OF,PROG1,8 to delete the program
<code>:</code>	directly from RTE.

Now go back to Step 2 and edit the version of your source program that is stored in SOURC2. This time you may want to save the new version when you are finished in SOURC1 so you have SOURC2 as a backup version. You do this by using the ERSOURC1 command when you have completed all editing.

Note that if you do not purge the PROG1 file as well as OF the program, you can still use the RUN command from FMGR to run PROG1. The File Manager RUN command will search for a file named PROG1 and initialize an ID segment for it automatically; you do not have to use the RP command. Therefore, if you want to use the program name PROG1 for your corrected program, you must be sure to purge the PROG1 file.

A SUMMARY

```
*RU,FMGR
:LS
:RU,EDITR
SOURCE FILE?
/Δ
EOF
/ΔFTN4,L
/Δ      PROGRAM PROG1
/Δ      DIMENSION NAME (10)
/Δ      DO 20 I=1,10
/Δ      NAME(I)=2HΔΔ
/Δ  20 CONTINUE
/Δ      WRITE (1,30)
/Δ  30 FORMAT ("PLEASE TYPE YOUR NAME AND PRESS RETURN.")
/Δ      READ(1,40) NAME
/Δ  40 FORMAT (10A2)
/Δ      WRITE (1,50) (NAME(I),I=1,10)
/Δ  50 FORMAT ("YOUR PROGRAM WORKS, ", 10A2)
/Δ      END
/Δ      END$
/1
      FTN4,L
/ECSOURC1
END OF EDIT
:LI,SOURC1                      (output is on line printer)
:RU,EDITR
SOURCE FILE?
/SOURC1
      FTN4,L
/12      50 FORMAT ("YOUR PROGRAM WORKS, ", 10A2)
/R      50 FORMAT ("GOOD. YOUR PROGRAM WORKS, ", 10A2)
/ECSOURC2
END OF EDIT
:LI,SOURC2                      (output is on line printer)
:RU,FTN4,SOURC2,LIST1,RELOC
:LG,2
:MR,RELOC
:RU,LOADR,99
/LOADR:PROG1 READY
/LOADR:$END
:RU,PROG1
PLEASE TYPE YOUR NAME AND PRESS RETURN.
KIMBERLY
GOOD. YOUR PROGRAM WORKS, KIMBERLY
:SP,PROG1
:EX
$END FMGR
```

RTE FEATURES

SECTION

III

In this section, additional features of RTE are discussed. Included here is an explanation of how BASIC runs in RTE, examples of batch jobs, and a brief discussion on the RTE Multi-Terminal Monitor. There's also an introduction on how to use transfer files. Using a transfer file, it's possible to compile, load, and run a program using one command. For more detailed information on any of these topics you should consult the appropriate reference manuals.

This manual discusses preparation of FORTRAN IV programs only. Should you want to program in a language other than FORTRAN IV, you will have to follow a different procedure during program development from the one that has been described here. It is recommended that you consult the RTE Reference Manual, as well as the appropriate language reference manual.

Multi-Terminal Monitor

The RTE Multi-Terminal Monitor allows multiple users to access the RTE system at the same time from different terminals. When you are using the system from a terminal other than the system console, you must be aware of the following:

- The prompt characters which RTE prints, when you press any key on your terminal to get the system's attention, are the logical unit number of the terminal followed by a "greater than" symbol, >. For example:

07>

After the prompt is printed you may enter commands as you do from the system console.

- If you want to use your terminal for input and output you must remember to use the LU number of your terminal instead of LU 1, the system console. For example if you want to change the list device from the standard one to your terminal you type:

:LL,7 (assuming your terminal is LU 7.)

Or if you use the File Manager STORE command to type data and store it in a file:

:ST,7,FILEX

- Error messages from the compilers, Assembler, and Loader are printed on the standard log device (usually the system console) and not at your terminal. If an error occurs that requires operator intervention, control may transfer to the system console. You must then correct the error at the console and type a colon (:) or TR to return control to your terminal.
- Since the system has at most one LG area and one LS area active, only one user should be using either of these areas at a time. If you are compiling, assembling, or loading from the LG area, you must make sure someone else is not trying to do this simultaneously. You should also avoid using the LS area for program development if another user is doing the same thing. You may do some program development simultaneously with other users by:
 - using EDITR to edit existing source programs from a file back to a file, avoiding the LS area,
 - creating a source program file with the File Manager STORE command, or
 - preparing a Batch Job which can be run under the Batch-Spool Monitor spooling process. (See the next topic for more information about spooling jobs.)
- It is recommended that the system have a different name for programs such as EDITR, FMGR, and BASIC for each terminal so more than one person can run a particular program at the same time. This is usually set up by the person who generates the system as a special procedure file (transfer file) which executes when the system is loaded. An example of such a file and the method of defining other names for these programs are described in Appendix B.

The program names usually incorporate the terminal LU number, for example, FMG07 and ED07. All other names for FMGR must begin with the letters FM. Ask the system manager if the program names are already set up on your system.

Figure 3-9 contains an example of preparing a program from a terminal which is LU 7.

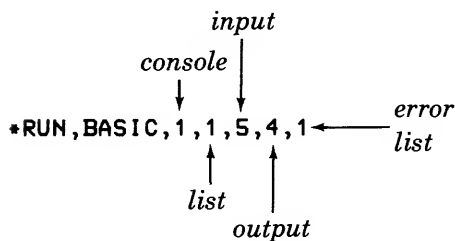
07>LS	◀ MTM prompts for a command. Clear the LS pointer.
07>EDI07	◀ Schedule EDITR for your terminal, EDI07.
SOURCE FILE?	
/	
EOF	
/ FTN4,L	
/ PROGRAM PROG1	
/ WRITE (7,100)	◀ Enter source program.
/ 100 FORMAT ("HI")	
/ END	
/ END\$	
/EC SOURC1	◀ Create file SOURC1.
END OF EDIT	
07>RU,FTN4,SOURC1,LIST1,RELOC	◀ Run compiler, which should be run by only one person at a time.
07>RU,FMG07	◀ Run File Manager from your terminal
:LG,2	◀ Declare LG size of 2 tracks.
:MR,RELOC	◀ Move RELOC file to LG area of disc.
:RU,LOADR,99	◀ Run Loader, which also should be run by only one person at a time.
:RU,PROG1	◀ Run your program, PROG1.
HI	◀ Output from PROG1.
:EX	◀ Terminate FMG07.

Figure 3-9. Using a Multi-Terminal Device

Basic Programs

Programs written in the BASIC language are executed under control of the BASIC Interpreter. They are not compiled or assembled like programs in the other languages. Complete information about BASIC commands and program statements is given in the Multi-User Real-Time BASIC Reference Manual. Therefore, this manual limits its discussion to accessing BASIC from RTE and the relationship of BASIC programs to File Manager files.

You can schedule BASIC with the RTE or FMGR RUN command. When you type RUN,BASIC, five parameters are set to default values are shown:



BASIC accepts commands and program statements from the LU specified as the *console*, lists programs to the *list* LU (for BASIC, this is the console, not the line printer), and lists errors on the LU specified as the *error list* parameter. The *input* parameter defines the source of input to your BASIC program; *output* defines the device for output from it.

BASIC prompts for commands and program statements with the "greater than" symbol, `>`. If your program calls any external subroutines you must enter the TABLES command first to declare the names of two tables used by BASIC, the Branch and Mnemonic Tables. The system manager will be able to tell you the names of these tables which are created with a table generator program, RTETG. For a description of the content and purpose of these tables see the Multi-User Real-Time BASIC Reference Manual. You do not need to know any more than the table names to create and run a BASIC program.

BASIC programs can be stored in File Manager files with the BASIC SAVE or CSAVE command and reloaded with the LOAD and RUN commands. The CSAVE command saves a semi-compiled form of the program. The LOAD command does not execute the program but merely makes it available for editing with the BASIC Interpreter. The RUN command runs the current program in memory if you do not provide a file name or loads and runs the program from a specified file. In BASIC the file name and the program name are the same.

Figure 3-7 illustrates one method for creating, saving and running a BASIC program. You can also create programs with the File Manager or EDITR but the advantage of creating them with the BASIC Interpreter is that your statements are checked for correct syntax when they are entered. Figure 3-8 contains an example of creating a BASIC program with EDITR.

The program in Figure 3-7 contains a subroutine call to an external subroutine, TIME. so the TABLES command must be used.

*RU,BASIC	◀ <i>Schedule BASIC.</i>
BASIC READY	◀ <i>BASIC indicates it is ready for a command or statement.</i>
>TABLES BRT,MNT	◀ <i>Declare the BRT and MNT Tables</i>
>100 X = 987 * 123	
>105 PRINT X	
>110 CALL TIME (X)	◀ <i>Type program statements.</i>
>115 PRINT X; " SECONDS AFTER MIDNIGHT"	
>120 END	
>SAVE PROGX	◀ <i>Create file named PROGX and save program in it.</i>
BASIC READY	
>RUN	◀ <i>Run current program.</i>
121401.	◀ <i>Print statement executed. X=121401</i>
50400.1 SECONDS AFTER MIDNIGHT	◀ <i>Subroutine call and print statement executed.</i>
BASIC READY	
>DEL	◀ <i>Type DEL to delete current program.</i>
>RUN	◀ <i>Note: If you now type RUN, there are no statements to execute.</i>
BASIC READY	
>RUN PROGX	◀ <i>If you type RUN with program name, PROGX is loaded and executed.</i>
121401.	
50430.2 SECONDS AFTER MIDNIGHT	
BASIC READY	◀ <i>BASIC waits for next command.</i>
>BYE	◀ <i>Type BYE to exit and return to RTE.</i>

Figure 3-7. Creating a Program Using BASIC Interpreter

*LS	◀ <i>Clear LS Pointer.</i>
*RU,EDITR	◀ <i>Run EDITR.</i>
SOURCE FILE?	
/Δ	◀ <i>Type space to indicate you are using LS area. EOF indicates no lines in EDITR work area.</i>
EOF	
/Δ100 X = 987 * 123	
/Δ105 PRINT X	◀ <i>Type space command, Δ, and new source statement.</i>
/Δ110 END	
/ECPROGZ	◀ <i>Type EC command and file name, PROGZ</i>
END OF EDIT	◀ <i>EDITR terminates.</i>
 *RU,BASIC	◀ <i>Run BASIC.</i>
BASIC READY	
>RUN PROGZ	◀ <i>Load and run PROGZ</i>
121401.	◀ <i>PROGZ prints value of X.</i>
BASIC READY	◀ <i>BASIC waits for next command or statement.</i>
>	

Figure 3-8. Creating a BASIC Program with EDITR

Batch Job Control

The sequence of commands you have been using to create, compile, load and run your program is comparable to a job stream. FMGR has two commands, JOB and EO, which allow you to prepare a job off-line on mini-cartridge or cards and then enter the job stream through a mini-cartridge or card reader instead of the system console or a terminal. More than one job can be entered in this way by using the job delimiters JO and EO. This method of entering commands is described in the Batch-Spool Monitor Reference Manual. The example in figure 3-10 will give you an idea of how to prepare a job.

The JOB command defines the beginning of the job and the EO command the end. You punch the sequence of commands on cards or tape in the same order you would type them on the system console except you must precede each command with a colon.

To enter the job, you use the RUN, FMGR command. First put the mini-cartridge or cards in the appropriate device and make sure the device is ready to read. Then type the RUN,FMGR command and specify as the input parameter the logical unit number of the device from which you are entering the commands. When the job is complete, exit from File Manager.

For example if the input device is a card reader defined as LU 10:

*RU ,FMGR ,10 Load the cards illustrated in figure 3-10. After the job has terminated, type EX to terminate FMGR. JO prints beginning of job message and EO prints end of job message on the line printer. For example:

```
JOB MYJOB      ON AT 08:32:30.73  ON 01 APR  1975

      .
      .
      .
      listings
      .
      .
      .

JOB MYJOB      OFF AT 08:32:41.50  ON 01 APR  1975
EXECUTION TIME: 00:00:27.68
```

The Batch-Spool Monitor provides you with other commands which allow you to set time limits to a job, switch logical units for the duration of a job, and print messages on the job list device. You can also store a job stream in a file and transfer control to the job stream when you want to execute it. You may use variable parameters when you define the job stream and then provide a value for the variables when you run the job. In this way you can create generalized procedure files (transfer files) for doing routine jobs such as program preparation.

Figure 3-11 illustrates how you might store a job in a file.

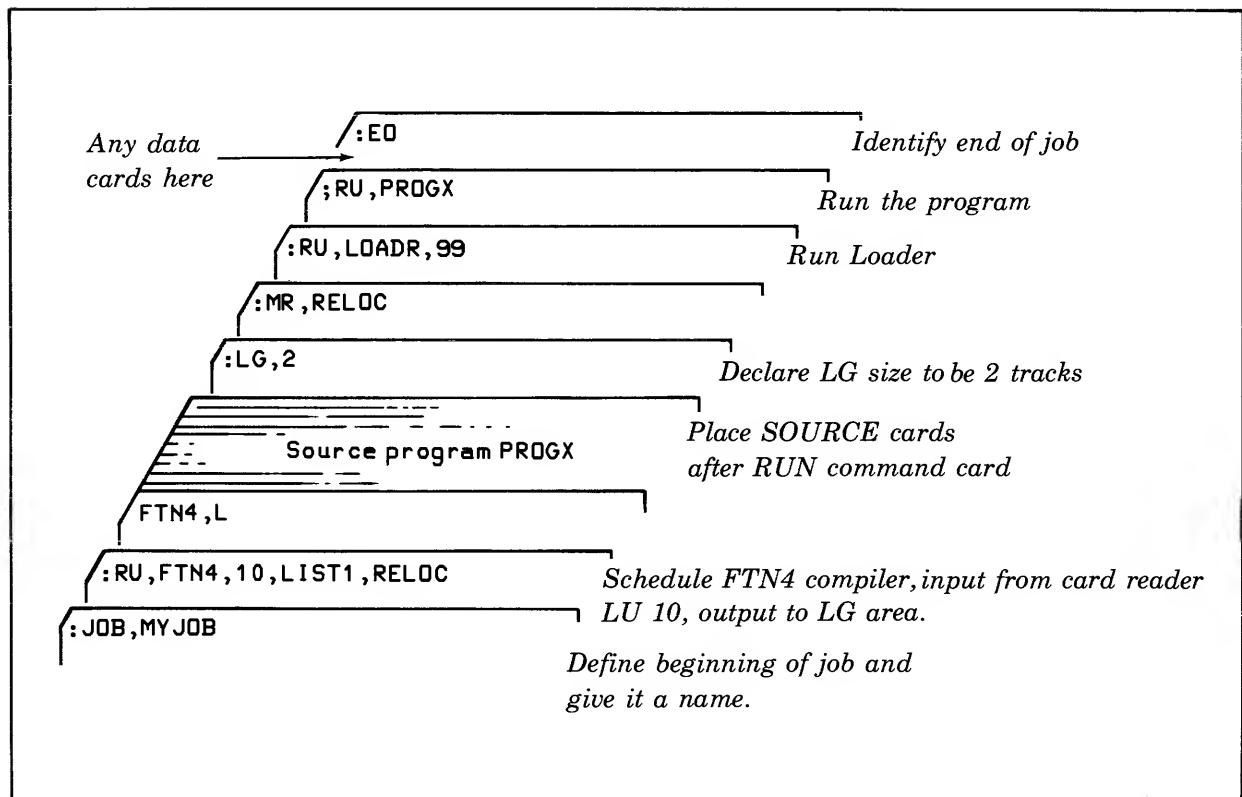


Figure 3-10. Job Deck Containing FMGR Commands

*RUN,FMGR	◀ Run FMGR.
:ST,1,MYJOB	◀ Store commands from LU 1, the system console.
:JOB,MYJOB	◀ Type first command. Remember, in this case you type the colon.
:RUN,FTN4,SOURC2,LIST1,RELOC	◀ Run FTN4 compiler.
:LG,2	◀ Declare LG size of 2 tracks.
:MR,RELOC	◀ Move RELOC file to LG area.
:RU,LOADR,99	
:RU,PROGX	
:EO	
Ⓚ	◀ Type Control D.
:	

Figure 3-11. Preparing a Procedure File Containing a Job

You can run the job by typing the TR command and the name of your job procedure file. For example:

<pre>:TR,MYJOB :JOB,MYJOB :RU,FTN4,SOURC1,LIST1,RELOC /FTN4: \$END :LG,2 :MR,RELOC :RU,LOADR,99 /LOADR:PROGX READY /LOADR: \$END :RU,PROGX :EO :</pre>	<p>Use the File Manager TR command to execute your procedure file. FMGR prints each command in the file as it is executed and the compiler and Loader messages are also printed.</p> <p>After the job is finished, FMGR prompts for the next command.</p>
--	---

You may also schedule the spooling program, JOB, and your job will run under control of the Spool Monitor. The advantage of executing your job this way is that you avoid conflicts with other users who want to share the LG and LS areas if all users prepare programs by this method instead of using copies of FMGR. There are two methods for requesting a spooled job. The first is:

<pre>*RU,JOB,MY,JO,B,10</pre>	<p>Schedule the program named JOB. The next three parameters define the name of the file containing your job. Specify the file name as pairs of characters separated by commas. The next parameter is the priority which is a number from 1 (highest) to 9999 (lowest and the default value).</p>
-------------------------------	---

The second method is:

<pre>*RU,JOB,1 ;:XEC,MYJOB,10 ; D^c</pre>	<p>Schedule JOB and type a 1 for the second parameter. JOB prompts for a command with a semi-colon (;). Type XEC and your file name separated by a comma. JOB prompts again. Reply with Control D. Your job is executed according to your priority and when the resources you want to use are free. The commands and other messages from the compiler and Loader are printed as shown with the TR command above.</p>
---	--

If you get this message:

```
END JOB ABNORM
RN02 nnnnn
JOB ABORTED
```

the Spool Monitor has not been initialized on your system. See the Batch-Spool Monitor Reference Manual for instructions on the initialization process or ask the system manager for help.

You may schedule a job from the system console or a MTM terminal. If the JOB program is busy, you will get the message, ILLEGAL STATUS. Try again since the JOB program executes so quickly that it will probably be free by the time you finish your request.

Error and Other Messages

Table 3-2 contains a summary of the termination and error messages from each compiler and the assembler. Error messages relating to source program errors detected during compilation are described in the language manuals.

The FORTRAN IV error messages are embedded in the following statement:

***program name** ERROR n DETECTED AT COLUMN cc*

and are printed on the standard list device instead of the system console.

The FORTRAN messages are printed in the following format:

E-nnnn: statement label+ ordinal line number

where the *ordinal line number* refers to the statement in which the error was detected. The line count does not include comment statements.

Assembler errors which occur during pass one of the assembly include the number of the tape where the error occurs, printed as *#n*. Errors which occur during the second pass include the page number on which the error occurs, PG *n*.

The following RTE system errors may occur when you are creating a program in any language:

Table 3-2. Termination and Error Messages

FORTTRAN IV	FORTTRAN	ALGOL	ASSEMBLER	MEANING
/FTN4: \$END	\$END FTN	\$END ALGOL	\$END ASMB \$END ASMB PASS	End of compilation. End of first assembly pass. Load source again. (Assembler only)
ERROR 01	E-0004	HPAL??	\$END ASMB CS	Missing control statement
ERROR 02	E-0004	HPAL??	\$END ASMB CS	Error in control statement
ERROR 03	E-0014		\$END ASMB SO	Symbol Table overflow
ERROR 06			\$END ASMB XEND	Missing \$ or END\$ statement (FORTRAN) or END statement (Assembler)

The following RTE system errors may occur when you are creating a program in any language.

<u>MESSAGE</u>	<u>MEANING</u>	<u>ACTION</u>
I/O ERR ET EQT # <i>eqt</i>	End-of-tape detected before an end of source program indication.	Put in ext source tape and type *UP, <i>eqt</i> .
IO06 <i>program name nnnnn</i> ABEND <i>program name</i> ABORTED	LG area not defined.	Declare LG area and start again.
IO09 <i>program name nnnnn</i> ABEND <i>program name</i> ABORTED	Overflow of LG area.	Declare a larger LG area and start again.

TROUBLESHOOTING

SECTION

IV

The purpose of this section is to help you determine what is happening with the system when you have initiated some action and the results are not what you expected or the action you requested is not taking place.

A characteristic of RTE which new users sometimes find disconcerting is the multi-programming environment. In this environment, you can schedule many programs to execute simultaneously rather than one at a time. The programs have priorities and the system determines which program uses which system resource (and when) on the basis of the priority assigned to the program. The priority is assigned to programs such as FMGR and EDITR when the system is generated.

Since scheduling a program is not the same as executing it immediately and programs are sharing the system resources, there are times when you think the system has ignored your request. Not so . . . the system is either carrying out your request or has recorded it to be taken care of as soon as possible.

How do you find out what the system is doing? Try using the tools described in this section:

- Use the STATUS command to determine the status of your program.
- Run the program named WHZAT to find out more about the status of your program, the status of I/O devices, and other programs running on the system.
- Check the input/output devices you are using and make sure they are operating properly.
- Look for some of the symptoms described in this section and follow the instructions provided to remedy the situation.

Program Status

You have had some experience with the STATUS command in Section I of this manual but now you will see its real utility. Type the command and the name of the program you are running.

*STATUS,FMGR

90 0 0 0 0 0 0 0

In this example, the FMGR program is dormant as indicated by the second number listed (the status) which is equal to 0.

The other status codes are:

- 1 the program is scheduled but is not executing yet,
- 2 the program is suspended and waiting for an I/O operation to complete,

- 3 the program is in a "general wait" state; that is, it is suspended for one of several reasons explained in the discussion of WHZAT which follows,
- 4 the program is suspended and waiting for the memory it requires to become available,
- 5 the program is suspended because it cannot get the disc space it requires,
- 6 an operator command or another program has suspended the program,
- 9 the program is a segment and has no status independent of the main program it belongs to.

Table 4-1 summarizes the action you may take if your program is in a particular state. If it is in states 1, 2, or 3, you will find it helpful to run WHZAT. If the program is dormant (not scheduled), it may have run to completion. If it is your own program and there are bugs in the code it may run to completion without doing what you think it should do.

Table 4-1. Programs States

PROGRAM STATE	ACTION
0 Dormant	Schedule the program again if you want to run it.
1 Scheduled	Terminate the program if you are the person who scheduled it. Try it again making sure all parameters are accurate and all devices it uses are operating and ready.
2 I/O Suspended	Check the device to see that it is ready for use. Sometimes you will have to wait for another program to finish using the device before your program gets access to it.
3 General Wait	Either continue to wait or make the resource the program is waiting for available. You may have to terminate another program. Be sure to only terminate your own or ask before terminating anyone else's program.
4 Unavailable Memory Suspend	Consult the system manager about getting more available memory.
5 Disc Allocation Suspend	Ask the system manager why there is no disc available. You can try releasing tracks from your own programs or the EDITR program with the File Manager RT command.
6 Operator or Programmed Suspend	Try typing the GO command to continue execution.

The WHZAT Program

WHZAT describes the current system environment. It operates in one of two modes which you select when you type the RUN command. One mode displays all scheduled and suspended programs and their status and the other (RTE-III only) displays in numeric sequence all partitions with their status.

The parameters for running WHZAT are:

**RUN,WHZAT,lu,option*

Where:

- lu* the logical unit number of the device on which you want the information displayed. The default value is LU 1, the system console.
- option* indicates what you want displayed.
0 (the default value) will cause the status of scheduled and suspended programs to be displayed.
- A non-zero value will cause the status of all partitions being used to be displayed (RTE-III only).

Both parameters are optional. If you do not supply them, the status of scheduled and suspended programs is displayed on the system console.

In either mode, the information is preceded by a heading which includes the current system time and column headings. Figure 4-1 contains a sample printout of the program status mode and figure 4-2 shows a sample of the partition status mode.

When you run WHZAT, your program may be listed in one of the following states:

- 0 program is dormant
- 1 program is scheduled
- 2 program is I/O suspended
- 3 program is suspended in general wait list
- 4 program is suspended waiting for system available memory
- 5 program is suspended waiting for disc tracks
- 6 program is either operator or program suspended

WHZAT provides information describing the following situations that occur when your program is in state 2 or 3.

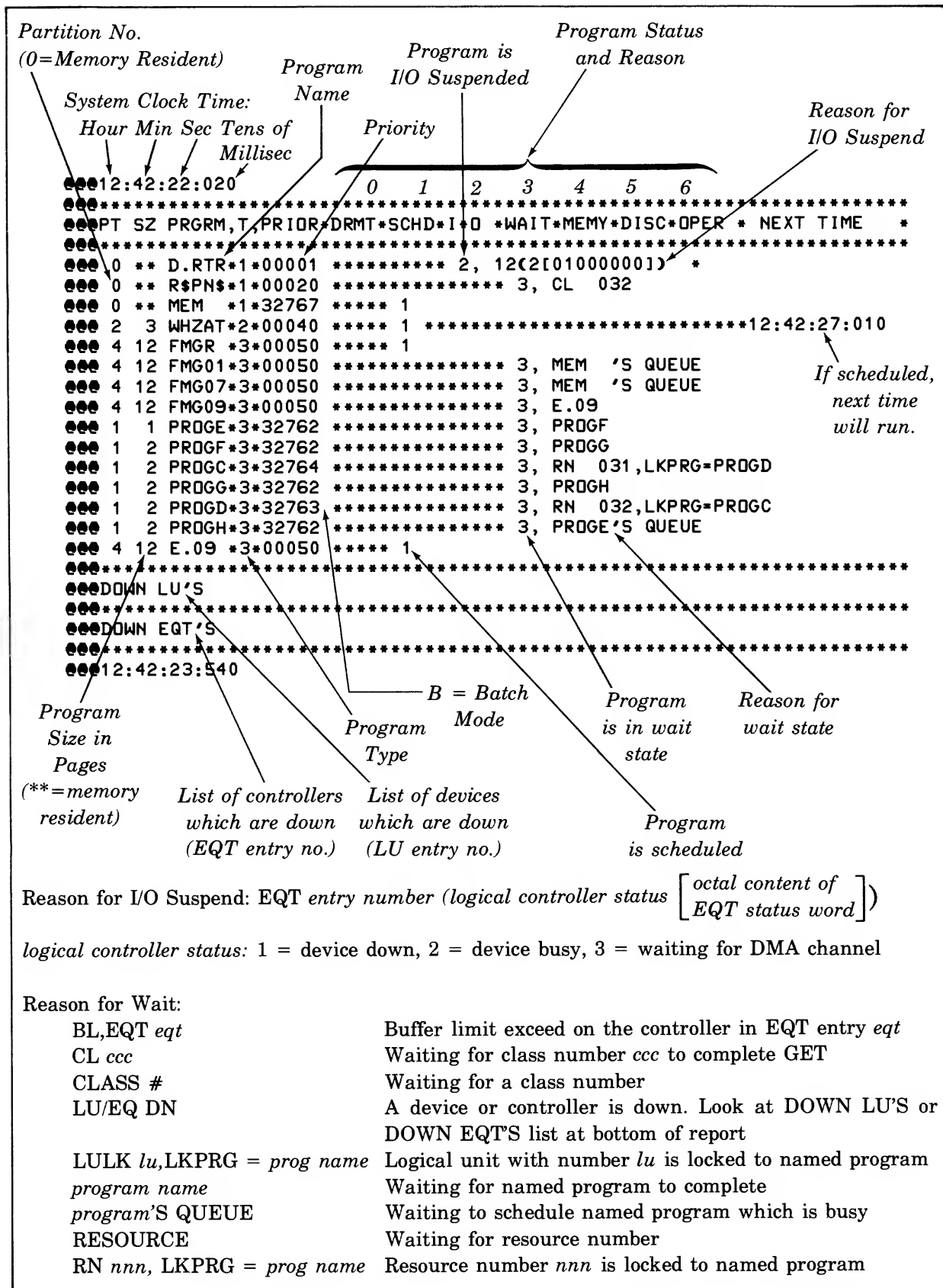


Figure 4-1. Program Status Mode (*RUN,WHZAT)

Your program is in the WAIT state (3) because an input/output device or controller is down and RTE cannot initiate an I/O operation requested by your program. WHZAT lists the program in state #3 with LU/EQ DN. Examine the list of DOWN LU'S or EQT'S at the bottom of the program status report to see which devices or controllers are down. After you make the device or controller ready, use the *UP,eqt command to notify the system that it is now operating properly or use the *LU,lu,eqt command to move the suspended I/O to another device.

Your program is in the WAIT state (3) because a logical unit it wants to use is locked to another program, LULK lu, LKPRG = program name. You can look at the status of the program which is using the device and decide to wait until the program has finished with it or give up for now and terminate your program with *OFF, program name, 1. If the other program's status indicates that it is having problems with the device, you can notify the system manager or the person who is running the program so they can decide what to do. They may decide to terminate that program and then yours will continue executing.

Your program is in the WAIT state (3) and waiting for another program to complete or waiting to begin running another program which is busy. Check the status of the other programs and decide whether to wait or terminate your program and try again later.

You must be careful when you terminate programs with the 1 parameter. If someone is using FMGR to pack the disc and you terminate FMGR with a 1, you can destroy the disc content. You should always determine that no one else is using FMGR for this purpose or for running a batch operation before terminating it with 1.

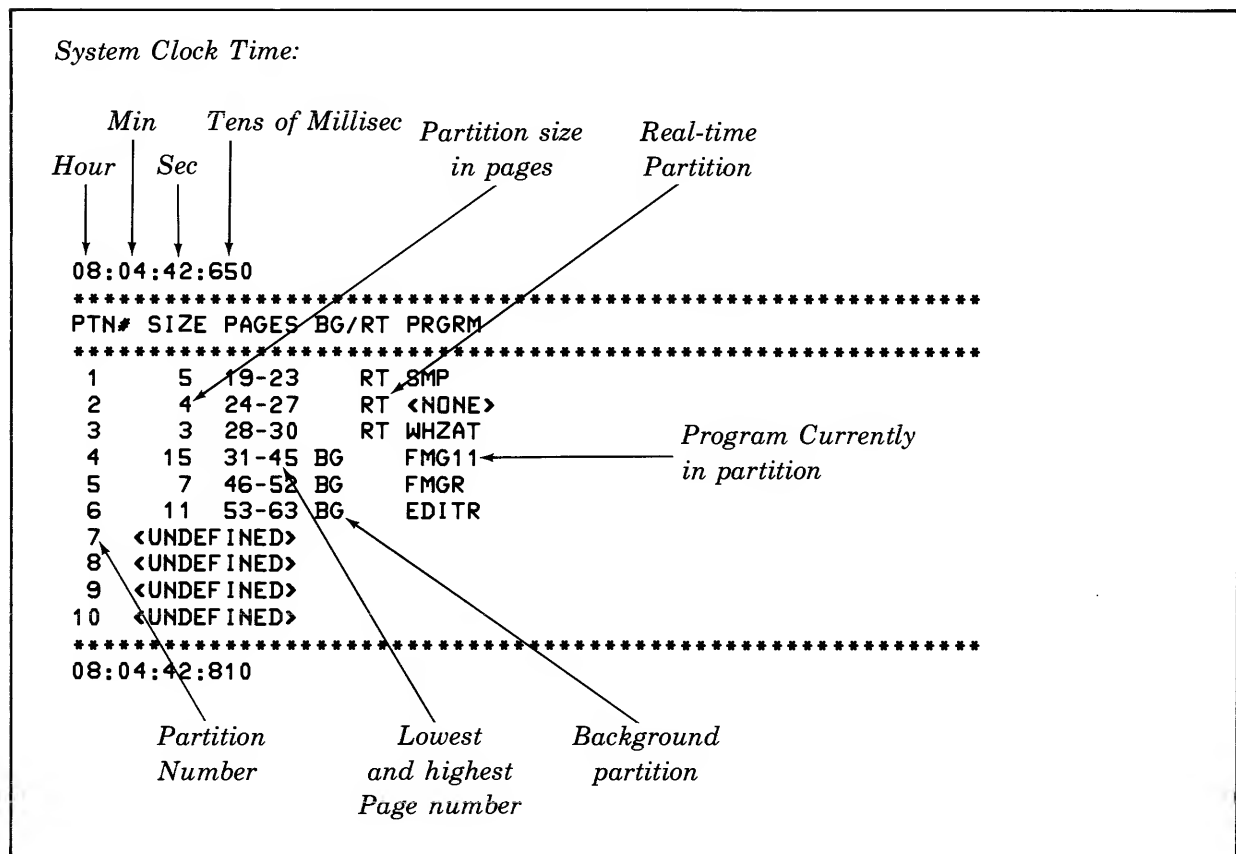


Figure 4-2. Partition Status Mode (*RUN,WHZAT,,1)

Your program is I/O suspended (state 2) because a device or controller is down. Check the DOWN LU'S or EQT'S list and correct the situation, then use the `*UP,eqt` command to notify RTE that the device or controller is ready. You can also use the LU command to move the I/O to a device that is up. (This situation is similar to the WAIT state EQ/LU DOWN but occurs when a device or controller goes down while you are using it rather than before I/O is initiated.)

Your program is I/O suspended (state 2) because a controller is busy. You must either wait for the controller to be free or terminate your program and try later.

Did you accidentally omit a parameter when typing a command so that the default logical unit number is not the device you really want to use? For example:

<code>*RU,LOADR</code>	The LOADR will try to read from the standard input device (LU 5) instead of the LG area since you did not specify the 99 parameter. This may cause your program to be I/O suspended.
------------------------	--

The partition status mode (RTE-III only) provides a dynamic map of the activity in each partition and the partition size.

Device Problems

Here is a checklist of things to do if you suspect device problems:

- Did you forget to use the UP command after getting a message that a device is down?
- Is each device set to the on-line or remote mode of operation?
- Does each device have its power on?
- Is the disc drive ready light on?
- Is the device physically connected to the system? Consult with the system manager.
- Did you load the paper tape backwards?

If you use the LL command and specify a logical unit number greater than 63, the system will translate the number you specify to the remainder of your number divided by 64. For example, if you type `*LL,79` the system will make the list device LU 15. If the device is not an appropriate list device the system may not carry out your request to list something. If you do not get a listing when you expect to, check the previous commands you have typed and see if you made a mistake.

Symptoms and Remedies

SYMPTOM	REASON	REMEDY
Unexpected * or 07> printed	You accidentally hit a key when RTE or the currently executing program was not waiting for input from the terminal.	Type an invalid command such as XX and press RETURN. An opcode error will be printed, ignore it and continue what you were doing before the * appeared.
IO09	LG tracks overflowed.	Declare larger LG area and begin again.
\$END ASMB CS \$END ASMB	You tried to assemble a FORTRAN program. This may happen if the current LS area contains a FORTRAN program and you did not declare a new LS area and move your Assembler code to it.	Declare a new LS area. Move source to it. Try Assembly again.
Unexpected message printed.	If you are working at the system console a message from another terminal's activity may be printed. If running FMGR and you give an RTE command, you may get FMGR message before RTE message. For example: :LI,SOURCX *STATUS,PROGX : <i>FMGR prompts for next command before</i> 0 <i>RTE prints the status.</i>	Ignore extra messages.
Lack of activity at your terminal.	You may have typed ΔRUN or ΔON at your terminal without giving your terminal number as the input device. The MTM will transfer activity to the default input device for the program you are running. (If it is FMGR, the default will be LU 1.)	Go to the console. Terminate the program and try again at your terminal.

SYMPTOM

REASON

REMEDY

You typed RUN,LOADR from File Manager and nothing seems to be happening.

The LOADR program is probably already scheduled.

Use the STATUS command or WHZAT to see if the version of FMGR you are using is in a General Wait state, 3. If so, you must wait until the other person is through using the Loader or if it has been left in a suspended state, you can use the OFF, LOADR,1 command to terminate it.

COPYING PROGRAMS

APPENDIX

A

To make a copy of a program and give it another name you must run FMGR. You use the SP (save program) command to create a type 6 file and save a copy of the memory image version of the program. Next you rename the program file using the RN (rename) command. Finally you create an ID segment for the new program by using the RP (restore program) command.

*RU,FMGR	Run File Manager.
:SP,FMGR	Save a copy of FMGR in a file.
:RN,FMGR,FMG07	Rename the file to FMG07 (use the number of your terminal). Make the
:RP,FMG07	file a temporary program by notifying RTE about the program with the
:EX	RP command.
\$END FMGR	Terminate the File Manager.

If you want to create several ID segments for accessing FMGR, one for each terminal, you need make only one copy of the program as a file. You can save this copy and then use the RN and RP commands to create multiple ID segments which reference the same type 6 file by different names. This is how you do it:

*RU,FMGR	Run File Manager.
:SP,FMGR	Save a copy of FMGR in a file.
:RN,FMGR,FMG07	Rename the file to FMG07.
:RP,FMG07	Create an ID segment pointing to the file.
:RN,FMG07,FMG09	Rename FMG07 (the file) as FMG09.
:RP,FMG09	Create an ID segment pointing to the file with program name FMG09.
:	Continue this pattern until you have created as many names as you want.
.	Then exit FMGR.
.	
.	
:EX	

Now the system console and two different terminals may use the File Manager program. Each terminal will call the program by a different name, FMGR, FMG07, and FMG09.

Use this method if you want multiple copies of a program and want to save disc space. Using the Batch-Spool Monitor you can create a procedure (transfer) file to set up these ID segments. It can be accessed each time the system is loaded. You will learn about this when you read the Batch-Spool Monitor Reference Manual.

Remember, names of copies of FMGR must start with letters FM.

To make a copy of BASIC using method 1:

*RU,FMGR	You can use five characters of your name for the copy of BASIC or some
:SP,BASIC	other name which is not already a program or file on the system.
:RN,BASIC,BASIX	You only have to do this once. Next time the system is loaded you can
:RP,BASIX	make a BASIX a temporary program with the File Manager RUN
:EX	command.

To make copies of the Editor using method 2:

*RU,FMGR	Create a copy for terminal 7.
:SP,EDITR	
:RN,EDITR,ED07	Create a copy for terminal 9.
:RP,ED07	
:RN,ED07,ED09	
:RP,ED09	
:	
	.
	.
	.
:EX	

To create a procedure file which will create multiple copies of a program, use FMGR to store the commands in a file. For example:

*RU,FMGR	Type the STORE command and store each command in a file named
:ST,1,NAMFP	NAMFP (in this example). Remember to type the colon preceding the
:SP,FMGR	command.
:RN,FMGR,FMG07	
:RP,FMG07	
:RN,FMG07,FMG09	
:RP,FMG09	
	.
	.
:RN,FMG14,FMGR	If you only use the procedure immediately after loading the system,
::	rename the file to the original file so it is ready the next time you use
Ⓚ	NAMFP. Type two colons.
:	Type Control D to indicate the end of file.
	FMGR prompts for the next command.

Whenever you want to execute the procedure, type the FMGR TR command.

- A command (abort), EDITR, 1-23
- ALGOL
 - input/output statement, 1-9
- ASMB (Assembler)
 - messages, termination and error, 4-7
- auxiliary disc, 1-9
- BACKSPACE key, 1-8, 1-10
- BASIC
 - commands, 3-4
 - copies, A-2
 - Interpreter, 3-4
 - programs, 3-4
- batch job, 3-6
 - deck, 3-7
 - processing, 1-13
- Batch-Spool Monitor, 1-13, 3-6, A-1
- bootstrap loader, 1-5
- Branch and Mnemonic Tables, 3-4
- CAP LOCK key, terminal, 1-8
- card reader, 1-9
- Cartridge Directory, 1-15, 1-16
- cartridge, File Manager, 1-15
 - label, 1-15
 - reference number, 1-15
- change list device, 1-18
- checklist, device problems, 4-6
- CL command (cartridge list), FMGR, 1-16
- CLEAR DISPLAY switch, 21MX, 1-5
- clock values, 1-11
- compile a program, 2-8
- control D, 1-17
- control H, 1-8, 1-10
- copying programs, A-1
- correct program logic errors, 2-15
- create a BASIC program, 3-5
- create a file, 1-16
- create a source file, 2-2
- CSAVE command, BASIC, 3-4
- current line, EDITR, 1-19
- D^c, 1-17
- declare LG area, 2-11
- delete lines, EDITR, 1-22
- DEL key, 1-8, 1-10
- developing a program, illustration, 2-14
- device busy, 4-5
- device down, 1-9
- device problems, 1-12, 4-6
- disc power, 1-2
- display register, 21MX, 1-4
- DL command (directory list), FMGR, 1-15
- documentation map, v
- DOWN EQT's, 4-3
- DRIVE READY light, disc, 1-3
- DUMP command, FMGR, 1-25
- EC command, EDITR, 1-23
- edit a source file, 2-6
- EDITR (Interactive Editor), 1-19, 2-2, 2-6, 2-14
 - copies, A-2
 - prompt character, 1-19
 - work areas, 1-20, 2-2
- EL,ELC,ELR commands, EDITR
- \$END ASMB CS, 4-7
- END JOB ABNORM, 3-8
- EO command (end of job), FMGR, 3-6
- EQT entry number, 1-12
- Equipment Table, 1-12
- ER command, EDITR, 1-21
- error messages, 1-23
 - FMGR, 1-14
 - MTM, 3-2
 - Step 1, 2-4
 - Step 2, 2-7
 - Step 4,
- *ERROR 01 DETECTED AT COLUMN 05, 4-7
- ERROR 05, 2-9
- EXEC calls, 1-9
- EXIT command, FMGR, 1-24, 3-6
- EXTEND light, 21MX, 1-4
- File Directory, 1-15
- File Management Package, iii
- File Manager, 1-13
- File Manager cartridge, 1-15
- FILE MANAGER ERROR-02, 2-7
- FILE MANAGER ERROR-06, 2-7
- FILE MANAGER-002, 2-4
- FILE MANAGER-006, 2-4
- FILE MANAGER-015, 2-4
- file name conventions, 1-14
- file operations, 1-13
- file types, 1-26
- fixed disc cartridge, 1-3
- FMGR, 1-13
 - error messages, 1-14
 - format conventions, 1-13
 - input device default, 1-13
 - prompt character, 1-13
 - RUN command, 1-13
- FMGR000 BREAK, 1-14
- FMGR010, 2-4, 2-7, 2-9
- FMGR014, 2-12
- FMGR050, 2-9, 2-12
- FMGR056, 2-4, 2-9
- FMGR-002, 1-24, 2-9
- FMGR-006, 1-24, 2-4, 2-7, 2-9
- FMGR-007, 1-24
- FMGR-008, 1-24
- FMGR-017, 2-4
- format conventions
 - FMGR commands, 1-13
 - RTE commands, 1-10

FORTRAN

- compile error, 4-7
- control statement, 2-2, 2-8
- input/output statements, 1-9
- program listing, 2-5, 2-10
- program statement, 2-2
- front panel, 21MX, 1-5, 1-6, 1-7
- FTN4 (FORTRAN IV compiler), 2-8, 2-14
 - RUN command parameters, 2-8
- general wait state, 4-2
- HALT switch, 21MX, 1-4
- HP 2645 Interactive Display Terminal, 1-8
- hyphen (-) command, EDITR, 1-22
- IBL switch, 21MX, 1-4
- I command (insert), EDITR, 1-21
- ID segment, 1-26, 1-27, 2-11, A-1
 - clear, 2-17
- ILLEGAL STATUS, 1-12, 2-4, 3-28
- INC M/DEC M switch, 21MX, 1-4
- input/output devices, 1-9
- Interactive Editor, 1-19
- INTERRUPT SYSTEM light, 21MX, 1-4
- IO01 FMGR, 2-7
- IO06 FTN4, 2-9
- IO07 FMGR, 2-4, 2-7
- IO09 FMGR, 4-6
- I/O ERR ET EQT #eqt, 1-12
- I/O ERR NR EQT #eqt, 1-12
- I/O ERR TO EQT #eqt, 1-12, 2-9
- I/O slots, 21MX, 1-4
- I/O suspend, 4-5
- JOB command, FMGR, 3-6
- job deck, 3-7
- JOB program, 3-8
- job stream, 3-6
- L command (list), EDITR, 1-20, 2-2
- LG pointer, 2-11
- line printer, 1-2, 1-9
 - turn on, 1-3
- LIST command, FMGR, 1-18, 2-2
- list
 - error messages, FMGR, 1-14
 - file content, 2-2
 - file names, 1-15
 - on system console, 1-18
- LL command, FMGR, 1-18 4-6
- load and run a program, 2-15
- LOAD command, BASIC, 3-4
- loading the RTE system software, 1-4
 - RTE-II, iii
- LOADR (Loader), 1-26, 1-27, 2-11, 2-14, 4-7
 - listing, 2-13
- /LOADR:DUPLICATE PROG NAME, 2-12
- /LOADR:L 08, 2-12
- /LOADR:L 10, 2-12
- LOAD/UNLOAD switch, 21MX, 1-3
- LOCAL/REMOTE terminal switch, 1-3
- locked logical unit, 4-3

I-2

LU, 1-9

LU0, LU8, 1-9

LULK lu, LKPRG=, 4-3

- magnetic tape drive, 1-9
- main power control, 1-2
- main program, 1-26
- memory image code, 1-26, 2-13
- minus (-) command, EDITR, 1-22
- Mnemonic Tables, Branch and, 3-4
- mount cartridge, 1-3
- MR command (move relocatable), FMGR, 2-8
- MTM (Multi-Terminal Monitor)
 - environment, 1-8, 1-13, 1-17, 2-1, 3-2, 3-8
- multiple program names, 3-2
- multi-programming environment, 4-1

N command (number), EDITR, 1-22

NO BLANK ID SEGMENTS, 2-12

NO SUCH PROG, 1-12, 1-27, 2-4

object code, 1-26

OFF command

FMGR, 1-25

program name, 1, 4-3

program name, 8, 1-23

RTE, 1-13

ON LINE/OFF LINE switch, line printer, 1-3

ON/OFF switch, terminal, 1-3

OP CODE ERR, 1-12

optional loaders, 21MX, 1-4

OVERFLOW light, 21MX, 1-4

P command (pending line), EDITR, 1-20

pack disc, 4-5

parameters, JOB program, 3-8

parameters, WHZAT, 4-3

PARITY LIGHT, 21MX, 1-4

partition, 1-11, 4-3

partition status mode, 4-3, 4-5

pending line, 1-19

command, EDITR, 1-20

permanent program, 1-27

plus command (+), EDITR, 1-20

power control button, 1-2

POWER ON switch, line printer, 1-3

prepare job off-line, 3-6

PRESET switch, 21MX, 1-4

priorities, program, 1-11, 4-1

procedure file, 3-2, 3-6, A-1

create, A-2

job, 3-8

prepare, 3-7

processor power, 1-2

program development, 1-26

program names, multiple, 3-2

program state, 1-11, 4-2

program status information, 1-27, 4-1, 4-2

prompt character

BASIC, 3-4

EDITR, 1-19

FMGR, 1-12

- JOB program, 3-8
- MTM, 3-2
- RTE, 1-10, 1-19
- PURGE command, FMGR, 1-24
- R command (replace), EDITR, 1-20
- rack (bay), 1-2
- real-time clock, 1-8
- rear panel, 1-3
- register select light, 21MX, 1-4
- release tracks, 1-24
- relocatable object (binary) code, 1-26
- removable disc cartridge, 1-3
- remove file from system, 1-24
- rename a file, 1-24
- renumbering process, EDITR, 1-22
- request program status, 1-11
- request time, 1-10
- RETURN key, terminal, 1-8, 1-9
- RNAME command (RN), FMGR, 1-24, A-1
- ROM bootstrap loader, 1-1
- RP command, FMGR, 1-27, A-1
- RTE commands, format conventions, 1-10
- RTE
 - prompt character, 1-10, 1-19
- RTE-II system, iii
- RTETG, 3-4
- RTRACKS command (RT), FMGR, 1-24
- RUBOUT key, 1-8, 1-10
- RUN switch, 21MX, 1-4
- RUN command
 - BASIC, 3-4
 - FMGR, 3-6
 - RTE, 1-10
- RUN/STOP switch, 1-3
- SAVE command
 - BASIC, 3-4
 - FMGR, 2-8
- schedule programs, 1-11
- segment, 1-26
- set pending line, EDITR, 1-20
- set real-time clock, 1-10
- set system on-line, 1-3
- SET TIME message, 1-8, 1-9
- size estimate LG area, 2-2
- slash command, EDITR, 1-20
- source file, 1-26
- space command, EDITR, 1-21, 2-2
- SP command (save program), FMGR, 1-26, 2-11
- spooled job, 3-8
- spooling, 1-13, 3-8
- Spool Monitor, 3-8
- standard input unit (device), 1-1, 1-9, 4-5
- standard list unit, 1-1, 1-9, 1-18
- standard output unit, 1-1, 1-9
- START switch, line printer, 1-3
- STATUS command, RTE, iii, 1-11, 4-1, 4-7
- STORE switch, 21MX, 1-4
- STORE command, FMGR, 1-16, 2-2, A-2
- subroutines, 1-26
- summary, program development, 2-16
- suspended program, 4-2
- symptoms and remedies, 4-7
- syntax errors, 2-15
- system console, 1-1, 1-2, 1-3, 1-8, 1-9
- system disc, 1-1, 1-9
- system library, iii
- system manager, 1-26, 2-13
- SYSTEM ON/OFF, 1-2
- system software, RTE, 1-3, 1-4
- Table Generator program, BASIC, 3-4
- temporary program, 1-27, 2-15
- terminal, 1-9
- terminate programs, 1-12
- TIME command, RTE, 1-10
- timeout value for a device, 1-9
- TM command, 1-10, 1-11
- TRANSFER command (TR), FMGR, 3-8
- transfer file, 3-24, 3-26, A-1
 - prepare, 3-7
- troubleshooting, 4-1
- turning on the equipment, instructions for, 1-2
- turn on disc, 1-3
- turn on system, 1-2
- two question marks, 1-14
- types of files, 1-26
 - types 3, 4, and 5, 1-26
 - type 6, 1-26, 3-10, A-1
- UNLOAD (STOP) position, 1-3
- UP command, 1-12, 4-3, 4-5
- WAIT state, 4-3
- WAITING FOR DISC SPACE, 2-12
- WHZAT program, 4-1, 4-3
 - partition status mode, 4-5
 - program status mode, 4-4
- XEC command (JOB), 3-8
- 21MX computer front panel, 1-4
 - illustration, 1-7
 - I/O slots, 1-4
 - ROM bootstrap loader, 1-4
- / command, EDITR, 1-20
- Δ command, EDITR, 1-21
- ?? command, FMGR, 1-13, 1-14, 2-4, 2-7
- + command, EDITR, 1-20
- command, EDITR, 1-22

READER COMMENT SHEET

RTE II/III: A Guide For New Users

92060-90012

AUG 1978

We welcome your evaluation of this manual. Your comments and suggestions help us improve our publications. Please use additional pages if necessary.

Is this manual technically accurate?

Is this manual complete?

Is this manual easy to read and use?

Other comments?

FROM:

Name _____

Company _____

Address _____

FOLD

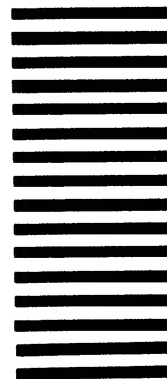
FOLD

BUSINESS REPLY MAIL

No Postage Necessary if Mailed in the United States Postage will be paid by

Hewlett-Packard Company
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014
ATTN: Technical Marketing Dept.

FIRST CLASS
PERMIT NO. 141
CUPERTINO
CALIFORNIA



FOLD

FOLD

PART NO. 92060-90012
Printed in U.S.A. 8/78



Sales and service from 172 offices in 65 countries.
11000 Wolfe Road Cupertino, California 95014